National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file *Votre référence*

Our file *Notre référence*

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

UNIVERSITY OF ALBERTA

Motion Tracking with a Pan/Tilt Camera

By

©

Don Murray

A thesis
submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree
of Masters of Science

Department of Electrical Engineering

Edmonton, Alberta
Fall 1992

National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-77118-6

Canada

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Don Murray

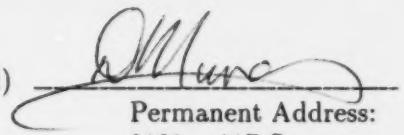
TITLE OF THESIS: Motion Tracking with a Pan/Tilt Camera

DEGREE: Masters of Science

YEAR THIS DEGREE GRANTED: 1992

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

(Signed) 

Permanent Address:
9131 - 117 St.,
Edmonton, Alberta,
Canada

Date: 6 October 1992

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled *Motion Tracking with a Pan/Tilt Camera* submitted by *Don Murray* in partial fulfillment of the requirements for the degree of Masters of Science.

V. G. Gourishankar

supervisor: V.G. Gourishankar

supervisor: A. Basu

R. E. Rink

examiner: R. Rink

external: H. Zhang (Computing Science)

Date: 5 October 1992

Abstract

This thesis describes a method for real-time motion detection from an active platform whose position is not accurately known. The camera is mounted on a pan/tilt device which provides rotation about two axes. Image mapping is used to align images of different viewpoints so that static camera motion detection can be applied. In the presence of camera position noise, the image mapping will be inexact and static camera methods will fail. The use of morphological filtering of motion images is explored to de-sensitize the detection algorithm to inaccuracies in background compensation. Two motion detection techniques are examined and experiments are run on stored image sequences to verify the methods presented. Experimental results are given and future improvements suggested. The system successfully extracts moving edges from dynamic image sequences taken with camera rotation about both pan and tilt axis.

Acknowledgements

I would like to thank both my supervisors; Dr. Gourishankar for his unfailing support, and Dr. Basu for his guidance and encouragement throughout the tribulations of this work. I am also grateful to Sergio Licardie and Darin Ingimarson for their ideas and friendship and the coffee we drank together, all of which supported and maintained me through the cruelest of motivation blocks. I would like to thank the memebers of my committee, Dr. Rink and Dr. Zhang for their valuable comments and criticisms. A special thanks goes to my father, Dr. Murray, for the many conversations we had on the nature of research, as well as for the many lunches he bought me and the proof-reading he endured. And, most of all, I would like to thank my fiancee, Gloria Chow, for her encouragement and inspiration, and for kicking me in the a_ when I needed it.

Contents

1 Introduction 1

1.1 Introduction to tracking 1

1.2 Inherent difficulties of computer vision 2

1.3 Thesis objective 3

1.4 Thesis organization 4

2 Tracking overview 5

2.1 Tracking methods 5

2.2 Optic flow tracking 6

2.2.1 Optic flow estimation from discrete images 7

2.3 Motion energy tracking 8

2.4 Camera rotation 9

2.5 Motion parameter extraction 10

3 System overview 12

3.1 System description 12

3.2 Modeling and Notation 13

3.2.1 Notation 13

3.2.2 Pin-hole camera model 14

3.2.3 Pan/Tilt Model 15

3.3 Justification of system design 17

3.3.1 Why not 3-D tracking? 17

3.3.2 Why active vision? 18

4 Background Compensation **27**
4.1 Derivation of compensation algorithm 29

5 Independent motion detection **34**
5.1 Introduction 34
5.2 Motion detection with a static camera 34
5.3 Motion detection by an active camera 37
5.4 Robust motion detection with an active camera 42
5.4.1 Morphological Filtering 43

6 Experimental Results **47**
6.1 Overview 47
6.2 Equipment layout 47
6.3 Experimental results 48
6.4 Discussion of results 48
6.5 Inaccuracies in moving edge detection 59

7 Analysis of compensation inaccuracy **61**
7.1 Introduction 61
7.2 Compensation error for pan-only rotation 62
7.3 Compensation error for pan and tilt rotations 64
7.4 Significance of error analysis 66
7.4.1 Maximum speed of tracking 66
7.4.2 Minimum speed of tracking 67
7.4.3 Filtering and sampling strategy 68

8 Conclusion **69**
8.1 Summary 69
8.2 Assessment 69
8.3 Future research 70

Bibliography **72**

A Projection of a sphere onto the image plane **75**

List of Figures

3.1 Pin-hole camera model 15
3.2 Pan/tilt device 16
3.3 Camera transformation with tilt 18
3.4 2-D centroid of the image 21
3.5 Projection of Sphere on Image Plane 23
3.6 Image of a Sphere 25
3.7 Error in Estimated 3-D Centroid Position 26

4.1 3-D point projected on two image planes 28
4.2 Camera transformations after pan and tilt rotations 29

5.1 Static camera sequence 35
5.2 Result of thresholded image subtraction and edge detection 36
5.3 Moving edges detected in frame 2 of static camera sequence . . . 37
5.4 Motion detection with a static camera 38
5.5 Motion detection with an active camera 39
5.6 Moving camera sequence 40
5.7 Image 1 of the moving camera sequence compensated for camera motion 40
5.8 Image subtraction with and without compensation 41
5.9 Example of mask for morphological filters 44
5.10 Subtracted Image with various sizes of erosion masks applied . . . 46

6.1 Pan-only image sequence 49
6.2 Moving edges in pan-only image sequence (Approach 1) 50

6.3 Moving edges in pan-only image sequence overlaid upon the originals (Approach 1) 51
6.4 Moving edges in pan-only image sequence (Approach 2) 52
6.5 Moving edges in pan-only image sequence overlaid upon the originals (Approach 2) 53
6.6 Pan and tilt image sequence 54
6.7 Moving edges in pan and tilt image sequence (Approach 1) 55
6.8 Moving edges in pan and tilt image sequence overlaid upon the originals (Approach 1) 56
6.9 Moving edges in pan and tilt image sequence (Approach 2) 57
6.10 Moving edges in pan and tilt image sequence overlaid upon the originals (Approach 2) 58

7.1 Magnitude of pixel mapping error 63

A.1 Imaging geometry of sphere 76

List of Tables

7.1 Pan-only compensation error 64
7.2 Worst-case compensation error 66

Chapter 1

Introduction

1.1 Introduction to tracking

Computer analysis of images is a rapidly expanding field with applications in diverse areas such as medical imaging, analysis of satellite photos and industrial quality control. Image processing is computer interpretation of 2-D discrete images which can represent visual, radar, X-ray, depth or infra-red information. Computer vision is a branch of image processing that deals with real-time analysis of a series of images to sense continual stimulus.

As computer vision matures, motion detection and tracking are becoming recognized as important capabilities in any vision system designed to operate in an uncontrolled environment. Animals excel in these areas, and that is because motion is inherently interesting and important. In any scene, motion represents the dynamic aspects. For animals, motion can mean danger or food, matters of life and death. For a mobile robotics platform, motion can imply a chance of collision, dangers to navigation, or alterations in previously mapped regions.

Tracking in computer vision, however, is still in the developmental stages and has had few applications in industry. It is hoped that tracking combined with other technologies can produce effective visual servoing for robotics in a changing work cell. For example, recognizing and tracking parts on a moving conveyor belt in a factory would allow robots to pick up the correct parts in a less stringent work atmosphere.

In this thesis, we will consider tracking with an *active camera. Active vision* implies computer vision implemented with a movable camera, which can intelligently alter the viewpoint so as to improve the performance of the system. An active camera tracking system could operate as an automatic cameraman for applications such as home video systems, surveillance and security, video-telephone systems or other tasks which are repetitive and tiring for a human.

1.2 Inherent difficulties of computer vision

Although computer vision has received significant attention from researchers over the last 25 years, the full potential of computer vision to provide general-purpose, practical, and real-time sensing is still far from being achieved. Why is this? One reason is that the range of capabilities conceivably possible to computer vision is enormous. We have merely to look at the biological evidence to see this. Many higher animals have sight as their prime sense for performing a multitude of tasks. These tasks include location and recognition of objects, navigation, control and manipulation of objects (hand-eye coordination), motion detection and tracking. The demonstrated capabilities of vision in the animal world provide the computer vision scientist with a never-ending series of challenging and difficult goals.

Clearly there is great potential in vision as a means of sensing. This is in part due to the great quantity of information available through visual stimulus. Yet the problems of computer vision often stem from poor quality of information, not lack of information. Although each image holds a large amount of information, much of it is irrelevant to the task at hand. As well, the information that *is* relevant is often difficult and expensive to extract. The difficulties of computer vision can be summarized into three categories: ill-posedness, ill-definedness and intractability [23].

A problem which is ill-posed, or underconstrained, does not have a unique solution. Since computer vision is the analysis of 2-D images constructed from three-dimensional scenes, there is necessarily a loss of information. Therefore, it is unrealistic to expect to reconstruct a 3-D model of an arbitrary scene from a single image. In the past the approach has often been to assume additional constraints to

the scene, such as smoothness. Such assumptions often decrease the robustness of the system. The increase in availability of sophisticated vision and robotics equipment has made multiple camera and moving camera systems feasible. Both of which have the potential for resolving this inherent underconstrainedness.

Ill-definedness refers to the difficulty in modeling surfaces of the scene. It is generally assumed that the surfaces viewed are smooth and Lambertian, which is not always the case in the real world. To robustly extract surfaces containing reflectance variation and non-gaussian noise, simple surface modeling is not sufficient. Ill-definedness extends into object modeling and recognition as well. The modeling of natural objects such as animals or trees is not well-defined and seems intangible to computer techniques.

Mathematical intractability in computer vision primarily refers to reverse-mapping and search problems. These problems are found when attempting to match parts of an idealized model to noisy images. Matching is an NP-complete problem. Its time complexity grows factorially with the number of parts in the model. If we have a noisy image, the initial segmentation is inevitably inexact. This means that we must also account for partial matches. Though a partial match can theoretically be reduced to a number of complete match problems, the actual time requirement is much higher and makes it impractical for a real-time system.

Even if the problem is 'tractable', operations in vision are inherently expensive. To this end, it is important to consider Rosenfeld's statement [23]: "In principle, the computations performed by a vision system should be chosen to yield maximal expected gain of information about the scene at minimal expected computational cost."

1.3 Thesis objective

The objective of this work is to design a method of real-time motion detection and tracking for an active camera system. The active camera system will be a pan/tilt arrangement and thus allow the camera to be rotated about two axes, giving is the ability to follow a moving object as it moves, and keep it within the

centre of the field-of-view of the camera. The experiments will be conducted with real images taken from a pan/tilt platform. However, the image processing will be done off-line. The emphasis is on the method of tracking, rather than actual real-time implementation. The definition of the problem is more completely stated in Section 3.1.

1.4 Thesis organization

The thesis is organized as follows. In Chapter 2 a brief overview of some of the topics which affect tracking, and in particular, tracking with an active camera.

Chapter 3 gives a description of the tracking system proposed in this thesis. The scientific notation used is presented and models for the camera and pan/tilt device are given. As well, a justification is presented for the design of our system attempting to show how the difficulties in Section 1.2 are overcome.

Our active camera is mounted on a pan/tilt device that allows rotation about two axes. Chapter 4 investigates the relationships between camera coordinate system positions and between pixel locations at different pan/tilt orientations.

Chapter 5 explains the methods of motion detection which were explored and developed.

In Chapter 6 the results of our motion detection algorithms are shown for processing performed off-line on stored image sequences. The advantages and disadvantages of the techniques are discussed.

Chapter 7 discusses some of the limitations of the system imposed by synchronization error and noise filtering. For given system and noise parameters, the upper and lower bounds of tracking velocity are examined.

The conclusions arrived at due to this work are presented in Chapter 8. As well, possible modifications and implementation issues are addressed.

Chapter 2

Tracking overview

2.1 Tracking methods

In general, there are two approaches to tracking which are fundamentally different, with different goals and methods. They are: recognition-based tracking and motion-based tracking.

Recognition-based tracking is really a modification of object recognition. If we can recognize a certain object in successive images, we can determine how it is moving. The advantage of this method of tracking is that it can be achieved in three dimensions. As well, the translation *and* rotation of the object can be estimated. The obvious disadvantage is that only a recognizable object can be tracked. Object recognition is a high-level operation which can be costly to perform. Thus, the performance of the tracking system is limited by the efficiency of the recognition method, as well as the types of objects recognizable. Examples of recognition-based systems can be found in the work by Lowe [19], Bray [10] and others [24] [11].

Motion-based tracking systems are significantly different from recognition-based systems. They rely entirely on motion detection to detect the moving object. They have the advantage of being able to track any moving object regardless of size or shape. Therefore, motion-based techniques are more suited for our system. Their disadvantage is that object orientation cannot be extracted. As well, 3-D tracking requires multiple camera systems or independent ranging

techniques.

Motion-based techniques can be further subdivided into optic flow tracking methods and motion-energy methods as described in Sections 2.2 and 2.3.

2.2 Optic flow tracking

In a non-static scene, for every image in an image sequence, we can attach an instantaneous *retinal velocity* to each point within that image. The field of retinal velocity is known as *optic flow* [5]. This field is effectively the 2-D velocity of every pixel within an image.

If we can extract this motion field for every image in an image sequence, and use it as the input to our motion analysis techniques, it is possible to determine *ego-motion* (camera motion) and detect independently moving objects.

The difficulty with optic-flow tracking is the extraction of the velocity field. By assuming the image intensity can be represented by a continuous function, $f(x, y, t)$, we can use Taylor series expansion to show that:

$$0 = \frac{\partial f}{\partial x}u + \frac{\partial f}{\partial y}v + \frac{\partial f}{\partial t} \tag{2.1}$$

where $u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$. Therefore, the instantaneous 2-D velocity of any point in the image is (u, v). This is a convenient equation since $\frac{\partial f}{\partial t}, \frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ all can be locally approximated.

The difficulty applying equation 2.1 is that we have two unknowns and only one equation. Thus, this equation describes a line on which (u, v) must lie, but we cannot solve it uniquely without additional constraints. One way to solve this problem is to employ relaxation techniques. This uses the assumption that the motion field is locally smooth, which may not hold for arbitrary scenes. This technique also requires iteration and several images in an image sequence, both of which are costly [13].

Some interesting work has been done with partial optic-flow information from an active camera image sequence to detect motion [21]. By moving a fully active camera (one which translates and rotates) with a known motion, rough estimates of the expected optic flow can be made. These estimates of motion

are qualitative. In other words, suppose the camera is translating to the left. The expected apparent motion of static objects will be to the right. The line upon which the motion should lie in the velocity space can be calculated from the motion of the camera. Any regions which are detected with significantly different lines of possible motion are concluded to be due to an independently moving object.

The significance of this work is that motion detection is achieved from a fully active camera. The algorithm can be evaluated quickly and hence we have a computationally efficient method for detecting motion from an unconstrained platform. The drawback is that since the evaluation is qualitative; it is less discriminatory than a quantitative approach. This means that objects with motion in similar directions as the apparent motion cannot be detected, whereas motion perpendicular to apparent motion will easily be detected. At present, this technique has only been used in motion detection without intelligent control of camera motion. However, it would be interesting to apply this work to active camera tracking in the future.

2.2.1 Optic flow estimation from discrete images

Since determining a complete optic flow field quantitatively is both expensive and ill-posed, for practical systems solving the problem for a few discrete points has been a popular alternative [14]. This method relies on identifying points of interest (also known as *features*) in a series of images and tracking their motion. The points of interest are selected by scanning the image for regions with high gradients in more than one direction. These points are indicative of corners in a 3-D scene [9]. The points of interest are identified in two consecutive scenes, and the motion of each point between images is measured to produce an estimated optic flow velocity for pixels within the neighbourhood of that point.

The disadvantage with this technique is that the points of interest in each scene must be matched to those of the previous image. This is difficult since such a matching problem is intractable in general. The problems increase in the case of an active camera. Since the scene viewed is dynamic, certain points will pass beyond the field-of-view while new ones will enter (drop-ins and drop-outs). This

entails a iterative search which must include the possibility that for each point there is no match. The complexity of this problem is such that it is not reasonable for real-time applications.

2.3 Motion energy tracking

Another method of motion detection is motion-energy detection. Motion-energy detection is a *spatio-temporal* method, since it involves simple filters in both the spatial and temporal domains. The basis of these methods is the temporal derivative. Let us consider a function $f(x, y, t)$ which describes the intensity of our input image. For a static scene with a stationary camera, without considering noise in irradiance or sensing, the derivative of this function with respect to time should remain zero. In other words, a pixel representing the same 3-D point, with constant illumination and reflectance, will have a constant greyscale value. If the pixel intensity changes dramatically, this can be due to motion. Either a new surface has occluded the previous one, or an occluding surface has been removed so that a different surface, with different reflection characteristics is now seen. Hence, by calculating the temporal derivative of an image and thresholding at a suitable level to filter out noise, we can segment an image into regions of motion and inactivity.

Although the temporal derivative is sometimes estimated by a more exact method, usually it is estimated by simple *image subtraction*:

$$\frac{df(x,y,t)}{dt} \approx \frac{f(x,y,t) - f(x,y,t-\delta t)}{\delta t}$$

This method of motion detection is subject to noise and yields imprecise values. Several schemes have been developed to improve the motion detection.

In general, techniques to improve image subtraction include spatial edge information to allow the extraction of moving edges, rather than regions of motion. Picton [22] utilized edge strength as a multiplier to the temporal derivative prior to thresholding. Allen [1] uses zero-crossings of second-derivative gaussian filtering as an edge locator, and combines this information with the local temporal and spatial derivatives and equation 2.1 to estimate the optic flow velocity of edge

pixels.

For practical, real-time implementations of motion detection, image subtraction combined with spatial information is the most widely used and successful motion detection method. In addition to computational simplicity, motion-energy detection is suitable for pipeline architectures which allow it to be readily implemented on most high-speed vision hardware. One disadvantage of this method is that pixel motion is detected but not quantified. Therefore, one cannot determine additional information such as the focus-of-expansion. Another disadvantage is that the techniques discussed are not suitable for application on active camera systems without modification. Since active camera systems can induce *apparent motion* on the scenes they view, compensation for this apparent motion must be made before motion-energy detection techniques can be used.

2.4 Camera rotation

Most work in computer vision has been done with stationary cameras. Of the few researchers who have worked with active vision, some apply a fully active camera (with both camera translation and rotation) [9] [21], while others constrain camera motion to only one form [2]. In this thesis, we will be considering the unique characteristics of an active camera with pan/tilt capability, i.e. one which can only rotate.

Theoretical work on this specific kind of camera motion has been done by Kanatani [16] [17] who has developed the fundamental geometry for such a system. From this work we know that, given a rotation of the camera by a rotational matrix R, a fixed 3-D point moves in the camera coordinate system as follows:

$$P' = R^T P \tag{2.2}$$

where P is the point location before rotation and P' is the location after rotation. This relationship between a 3-D point in camera frames at different orientations allows us to compensate for active camera rotation. This is crucial for our method of compensation for apparent background motion and will be dealt with specifically in Chapter 4.

2.5 Motion parameter extraction

The fundamental step in motion tracking must be motion detection. However, to track a moving object successfully, motion parameters must be extracted and used to predict future positions of the tracked object. Due to the delays in processing, position information will always be out-of-date. Therefore, if we move the camera to look at the detected position, the camera will always lag behind the immediate position of the object. By extracting the motion parameters we can predict the future position of the object and move the camera to intercept the anticipated position.

There has been a lot of work recently in use of Kalman filters for motion parameter estimation [18] [12]. While Kalman filters allow more adaptability to noise and target motion, they impose additional computational burdens on the system.

One approach to obtain the best compromise to this problem suggests using fixed-gain filters with gain values based on noise characteristics of the known tracking system [1]. Allen uses an $\alpha - \beta - \gamma$ filter with gains arrived at by Kalata [15] as optimal fixed gains derived from Kalman filter steady-state solutions. Kalman filters adapt to changing noise conditions. If the noise present in the system is constant, the Kalman filter adaptation will reach steady-state values for filter gains. By examining the filter-gain-to-noise relationships, the computationally expensive process of adaptation can be skipped by assuming these steady-state values from the onset.

In such a filter, position, velocity, and acceleration are estimated as shown in the following equations. In these equations x_e, v_e, and a_e are the estimated motion parameters (position, velocity and acceleration) using previous predictions and the current measured values; x_p, v_p, and a_p are the predicted motion parameters, which incorporate only prior information to extrapolate the current values; and x_m is the measured position, which is corrupted by noise. The noise in this case comes from two sources: position noise due to variance in acceleration of the target, and the variance of the noise in the measurement of the object's position from an image.

The predicted motion parameters are found using standard rectilinear motion [1]

$$\begin{aligned} x_p(t) &= x_e(t-1) + Tv_e(t-1) + \tfrac{1}{2}T^2a_e(t-1) \\ v_p(t) &= v_e(t-1) + Ta_e \\ a_p(t) &= a_e(t-1) \end{aligned}$$

where T is the time between two consecutive estimates. The updated motion parameter estimates determined by the $\alpha - \beta - \gamma$ filter are

$$\begin{aligned} x_e(t) &= x_p(t) + \alpha[x_m(t) - x_p(t)] \\ v_e(t) &= v_p(t) + \tfrac{1}{T}\beta[x_m(t) - x_p(t)] \\ a_e(t) &= a_p(t) + \tfrac{1}{2T^2}\gamma[x_m(t) - x_p(t)] \end{aligned}$$

in which the values of α, β and γ are constants determined by

$$\frac{\gamma^2}{4(1-\alpha)} = \lambda^2$$

$$\alpha = \sqrt{2\beta} - \frac{1}{2}\beta$$

$$\gamma = \frac{\beta^2}{\alpha}$$

and λ is determined from the known noise characteristics σ_a (position uncertainty due to variance of acceleration) and σ_n (the measurement noise variance) as

$$\lambda = \frac{T^2\sigma_a}{\sigma_n}$$

Allen notes that the values σ_a and σ_n are difficult to obtain and so he treats them as tuned parameters in the filter.

Such a filter is computationally efficient. Also, for constant noise the filter is robust and accurate since it is based on optimal Kalman filter solutions. The disadvantage of such a fixed-gain filter is that the assumption of constant noise is unlikely to be true in an unconstrained situation. For our applications, the assumption of constant σ_n may be valid. For tracking an unpredictable object such as a person or an animal, the assumption for constant σ_a will not be true. Hence this technique will not perform optimally. To achieve better estimates a fully adaptive filter must be used, at the expense of computational performance.

Chapter 3

System overview

3.1 System description

The objective of this thesis is to design an active camera tracking system. The hardware is comprised of a camera mounted on a pan/tilt device to allow two degrees-of-freedom in orientation. The change in orientation is achieved by rotation about two intersecting axes of the the pan/tilt device. The strategy is to maintain the camera Z-axis passing through the estimated centroid of the moving object. The position of the 3-D centroid of the object in spherical coordinates can be described by two angles and a distance. For our application, only a direction is necessary, and hence only two angles are estimated.

Tracking is achieved through the use of spatio-temporal filters for detecting motion energy. Since motion-energy detection is based on a static camera, compensation techniques for camera motion are developed. As well, additional filtering for motion detection is explored to improve the robustness of the system.

Motion parameters of the tracked object are estimated in terms of angular position, velocity and acceleration for each of the pan and tilt axes.

3.2 Modeling and Notation

3.2.1 Notation

Throughout this work, certain notation and conventions will be used. These are presented here.

Since we often discuss point locations in both two and three dimensions, it is important to differentiate between them. A location in 3-D is written symbolically in capital letters as (X, Y, Z) or is presented as a column vector P where:

$$P = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Two dimensional points are written in lower case such as: (x, y).

Arbitrary homogeneous transformations are formulated as a 4×4 matrix T where:

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \rho_x \\ r_{21} & r_{22} & r_{23} & \rho_y \\ r_{31} & r_{32} & r_{33} & \rho_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where the 3×3 sub-matrix $[r_{11} ... r_{33}]$ represents the orientation of the destination frame, and $[\rho_x, \rho_y, \rho_z]$ denotes the translation vector from the source to the destination frame.

Rotation matrices are special cases of arbitrary transformation where $\rho_x = \rho_y = \rho_z = 0$. In this work we use two types of rotations, rotation about the X-axis $[Rot_X(\theta)]$ and rotation about the Y-axis $[Rot_Y(\theta)]$. These can be expressed as

$$Rot_X(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rot_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A frame which undergoes a rotation is expressed as:

$$T_{NEW} = RT_{OLD} \tag{3.1}$$

Also, a 3-D point in one frame can be expressed in another frame by:

$$P_1 = T_2 P_2$$

where T_2 is the description of frame 2 with frame 1 as reference, and P_1, P_2 denote the point in frame 1 and frame 2 respectively.

For the pan/tilt and camera parameters:

f is the focal length of the camera

θ is the tilt angle from the level position

α is a small angle of rotation about the pan axis

γ is a small angle of rotation about the tilt axis

The word *frame* is used often in this work and can have the meaning of an image in an image sequence, as used by computer vision researchers, or as a coordinate system expressed by a homogeneous transform as used by robotics researchers.

3.2.2 Pin-hole camera model

Throughout this work, the pin-hole camera model is used. This model is the standard for single lens cameras. As shown in Figure 3.1, let $OXYZ$ be the camera coordinate system. The image plane is perpendicular to the Z-axis and intersects it at a point $(0, 0, f)$ where f is the focal length. Using this model,

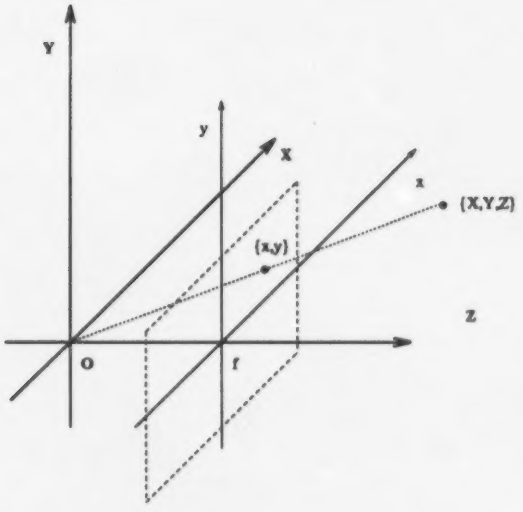


Figure 3.1: Pin-hole camera model

the relationships between points in the image plane, and points in the camera coordinate system are:

$$x = f\frac{X}{Z} \quad y = f\frac{Y}{Z} \tag{3.2}$$

where (X, Y, Z) is a point in the camera coordinate system, and (x, y) is the corresponding point in the image plane.

3.2.3 Pan/Tilt Model

The active camera considered in this work is mounted on a pan/tilt device that allows rotation about two axes. Figure 3.2 shows a drawing of the system used. The reference frame for each camera position is formed by the intersecting axes of rotation (pan = Y-axis, tilt = X-axis). The origin of the camera coordinate system is located at the lens centre which is related to the reference frame by a homogeneous transformation T_c such that:

$$P_r = T_c P_c \tag{3.3}$$

where P_r and P_c are 3-D points in the reference and camera frame respectively.

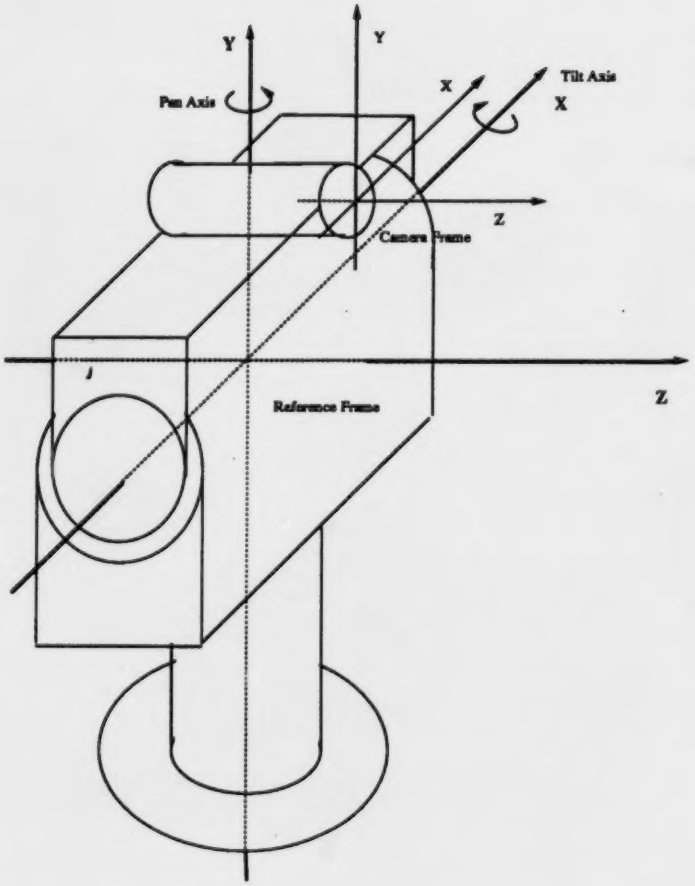


Figure 3.2: Pan/tilt device

From Figure 3.3 we can see that for an arbitrary tilt angle θ, the camera transformation T_c is:

$$\begin{aligned} T_c(\theta) &= Rot_X(\theta)T_c|_{\theta=0} \\ &= \begin{bmatrix} 1 & 0 & 0 & \rho_X \\ 0 & c\theta & -s\theta & c\theta\rho_Y - s\theta\rho_Z \\ 0 & s\theta & c\theta & s\theta\rho_Y + c\theta\rho_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{3.4}$$

Where:

$$T_c|_{\theta=0} = \begin{bmatrix} 1 & 0 & 0 & \rho_X \\ 0 & 1 & 0 & \rho_Y \\ 0 & 0 & 1 & \rho_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

(Please note the abbreviation of cos as c and sin as s.) This transformation for the camera position at arbitrary tilt angles is used in Chapter 4 in the derivation of the background compensation algorithm.

3.3 Justification of system design

3.3.1 Why not 3-D tracking?

In principle, tracking in 3-D is more desirable than tracking in 2-D, since it yields more complete information about the behavior of the tracked object. In this work, however, we only track in 2-D in a *spherical* space about the centre of the camera lens.

The reason for this simplification lies in our discussion of the three inherent difficulties of computer vision (see Section 1.2). Position extraction in 3-D from a single camera is an ill-posed problem. For an arbitrary object, there is no way to estimate depth from a 2-D image. To incorporate depth estimation, independent range-finding techniques, such as focus ranging, must be employed. A multiple camera system can extract range information. However, the inclusion of multiple input images greatly increases the computational cost of tracking, since the correspondence problem must be solved.

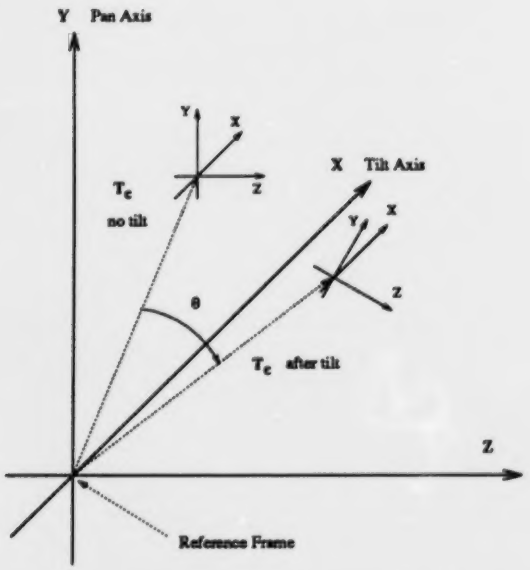


Figure 3.3: Camera transformation with tilt

Since our goal is to follow a moving object with a rotating camera, knowledge of depth would not significantly improve the tracking performance. In consideration of Rosenfeld's statement [23] about maximizing information gain with minimal computation cost, 2-D tracking seems to be the optimal method to employ.

3.3.2 Why active vision?

In the last 5 years, active vision has been receiving increasing support from the computer vision research community. It has been proposed that active vision can be beneficial in overcoming the problems of ill-posedness mentioned in Section 1.2 [3] [4].

Often a difficult vision problem can be simplified by looking at an object from the optimal viewpoint. However, determining the optimal viewpoint is also a difficult problem. By combining information from several viewpoints the amount and quality of information can be improved. A fixed camera system, by definition, must maintain a fixed viewpoint and consequently does not have the potential

that active vision has. While theoretical advantages of active vision are clear, in order to use active systems optimally, one must be able to intelligently select a viewpoint which yields a well-posed interpretation of a problem. Such intelligent behavior has yet to be fully explored.

For the tracking problem, certain specific advantages are obtained through use of an active camera. They are:

- increased field of view
- *foveation* (region of interest processing can be simplified)
- improved centroid estimation

These advantages are detailed below.

Field of View

For a general solution of the tracking problem, one would like as few artificial constraints in the system as possible. In an unconstrained situation, there is no limit on the location to which an object can move. To successfully track such an object, a very wide field-of-view is necessary. For a static camera to achieve a wide field-of-view, a wide-angle lens must be used. The effect of the wide-angle lens is to compress a large scene into a fixed image sensor area. As the field-of-view becomes larger, the resolution is reduced, since the sensor area remains the same. If details of an object are to be preserved, there is clearly a trade-off between resolution and field-of-view for a static camera.

An active camera, however, can overcome this trade-off. By moving the camera so that the object of interest remains in the camera's field-of-view, resolution reduction can be avoided. Also, a camera with an active zoom lens can adjust the scope of its field-of-view. Hence, if more detail is required, the camera can magnify the object. If the object is moving quickly and erratically, the zoom lens can be moved to minimum magnification, yielding a wider field-of-view with more robust tracking characteristics.

Foveation

A technique to overcome the field-of-view/resolution trade-off mentioned above is to model the camera after biological vision sensors such as the human eye. This method, known as *variable resolution* or *anthropomorphic vision* combines different resolutions in the same image [6] [25]. The central region, or fovea, has high resolution. This allows details to be obtained from selected objects viewed in the fovea. As we move to the edge of the field-of-view, however, the resolution is reduced. The purpose of this low resolution region is similar to human *peripheral vision*. At low resolution, a wide field-of-view can be covered at low computational cost. The high resolution fovea allows detailed processing of the centre, while the low resolution periphery allows coarse processing over a wide angle. The combination yields an over-all wide field-of-view without sacrificing detail of the object of interest. This is advantageous for such problems as navigation, character recognition [7], binocular camera vergence [8], and especially tracking. As with human vision, the peripheral vision allows fast motion detection over a large area. As the camera centres on the detected motion, more detailed motion estimation as well as other image processing operations can be made.

Although it is possible to apply variable resolution to a stationary camera system, the fovea would have to be moved about the image plane so as to capture the object of interest. Having a movable fovea in the image plane would make hardware implementation of variable resolution unfeasible and increase the computational cost. Therefore, to maintain the object of interest within the high-resolution fovea, an active camera is necessary. This application for active vision is very appealing since it closely imitates biological vision systems, which are remarkable in their flexibility and robustness.

Improved centroid estimation

Since we are only tracking the direction of the centre of the object, our position estimation is simplified to obtaining two parameters. The ray projecting to the centroid of the object is estimated to be the line from the lens centre through the centre of the image (see Figure 3.4).

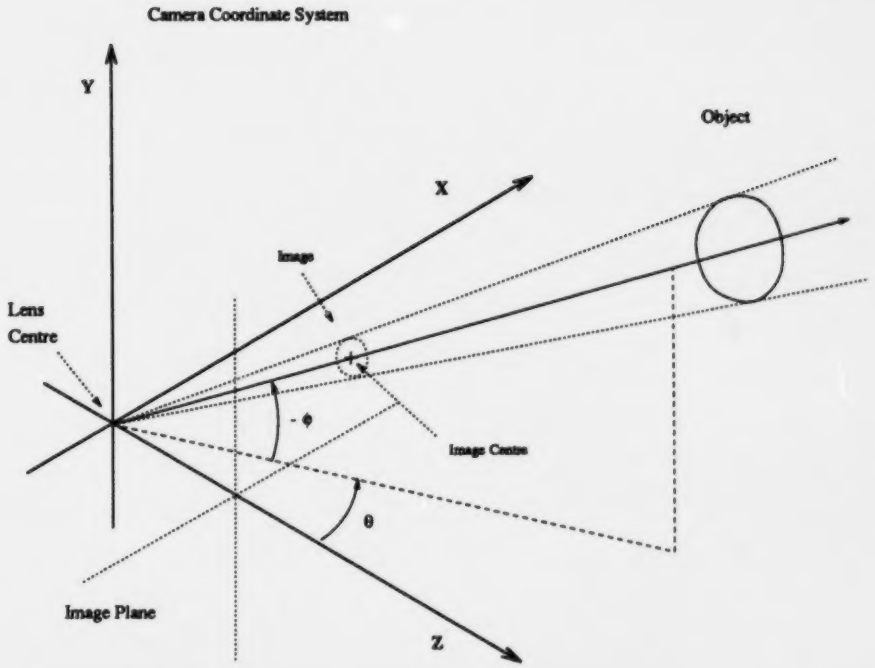


Figure 3.4: 2-D centroid of the image defined in polar coordinates, θ and ϕ are the estimated parameters

This method of estimating the position of the object is inherently inaccurate, since we have no *a priori* knowledge of the object's structure. However, if we remove as many known sources of error as possible the estimate of position can be improved. One source of error is the distortion of the image caused by the imaging process: *perspective projection.*

Determining the distortion of an arbitrary object is not practical since it is dependent upon the object's structure and orientation. Thus, we consider a sphere as an illustrative example. A sphere is a good sample object, since it is radially symmetric and simplifies the mathematics of the error analysis.

First of all, let us consider the image centroid versus the 3-D centroid of a group of points. Using the camera model mentioned in Section 3.2.2, consider n points $(X_i, Y_i, Z_i)\{i = 1, ..., n\}$. The three-dimensional centroid is:

$$X_c = \frac{1}{n}\sum_{i=1}^{n} X_i, \quad Y_c = \frac{1}{n}\sum_{i=1}^{n} Y_i, \quad Z_c = \frac{1}{n}\sum_{i=1}^{n} Z_i \tag{3.6}$$

Using the pin-hole camera model and equation (3.6), the three-dimensional centroid projected onto the image plane located at (x_c, y_c), is given by:

$$x_c = f\frac{X_c}{Z_c} = \frac{f}{nZ_c}\sum_{i=0}^{n} X_i, \quad y_c = f\frac{Y_c}{Z_c} = \frac{f}{nZ_c}\sum_{i=0}^{n} Y_i$$

Whereas, by similar triangles and the definition of a two-dimensional centroid, the centroid of the corresponding image points at (x_{ic}, y_{ic}) is:

$$x_{ic} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{f}{n}\sum_{i=1}^{n}\frac{X_i}{Z_i}, \quad y_{ic} = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{f}{n}\sum_{i=1}^{n}\frac{Y_i}{Z_i} \tag{3.7}$$

Note that x_{ic} and x_c (and likewise y_{ic} and y_c) are the same if the depth (Z) is constant for all points (i.e. $Z_i = Z_c$ for all i). This is true if all the points (X_i, Y_i, Z_i) are on a plane parallel to the image plane. In other words, if the object lies on a two-dimensional plane parallel to the image plane, one can obtain an undistorted image.

Let us now consider an illustrative sphere to see how the three-dimensional centroid is related to the centroid in the image plane. Let **S** be a sphere with radius r_s and centre at (X_s, Y_s, Z_s). From the symmetry of the geometry of the

camera model about the Z-axis, it is evident that we can rotate our coordinate system about the Z-axis with no change in the image shape. Therefore, without loss of generality, we select a particular case for which we constrain $Y_s = 0$. Hence the centre of the sphere **S** rests on the X-Z plane (see Figure 3.5).

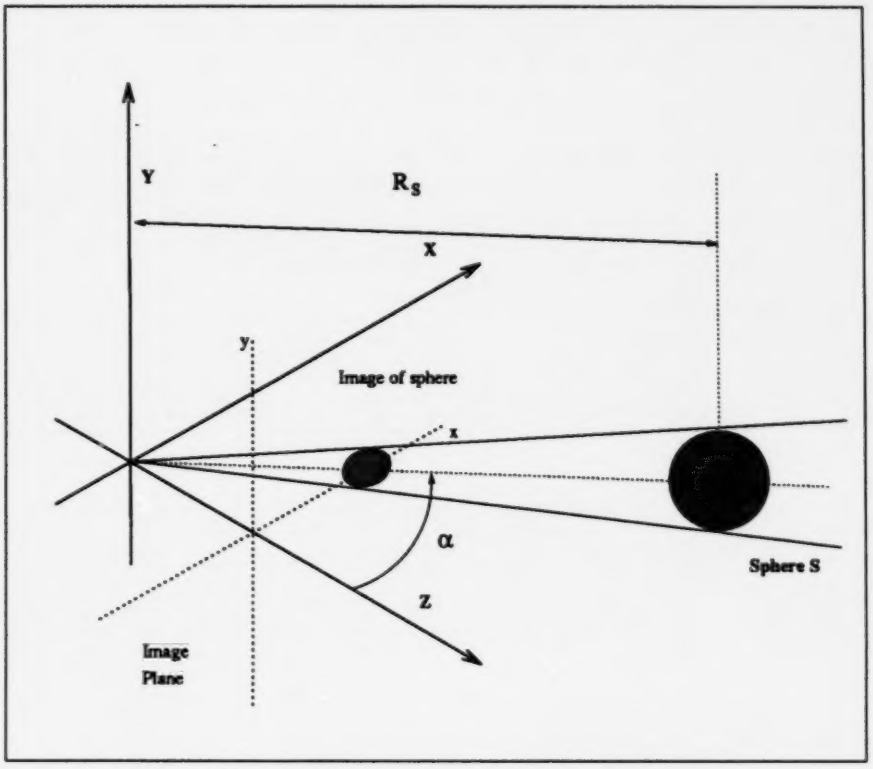


Figure 3.5: Projection of Sphere on Image Plane

In polar coordinates, the position $(X_s, 0, Z_s)$ can be replaced by an angle α from the Z-axis, and a range R_s. The boundary of the image of the sphere is formed by rays that pass through the lens centre and are tangential to the surface of the sphere. The tangential points exist in a single plane and form a circle. Hence, the image of a sphere is the same as the image of the circle thus described.

With this knowledge it can be shown (for details, see appendix A) that the

projection of the sphere on to the image plane is:

$$r_c^2 = R_c^2 \frac{[(x - \frac{f\,sin(\alpha)}{cos(\alpha)})^2 cos^2(2\alpha) cos^2(\alpha) + y^2]}{[f\,cos(\alpha) + x\,sin(\alpha)]^2} \tag{3.8}$$

where R_c is the distance to the centre of the occluding circle seen by the camera, and r_c is the radius of that circle.

If we consider the numerator of this equation, we see that it represents an ellipse, with centre at $(\frac{f\,sin(\alpha)}{cos(\alpha)}, 0)$. This centre is actually the same as the three-dimensional centroid projected on the image plane. The effect of the denominator, however, is to amplify the size of the ellipse as we move further away from the Y-Z plane. Thus, the shape tends to extend further away from the origin and pulls the image centroid in that direction. Note that this effect is accentuated by increasing α (Figure 3.6). In fact, for $\alpha = 0$,

$$\frac{r_c^2 f^2}{R_c^2} = x^2 + y^2$$

which is a circle centred at the origin with radius $\frac{r_c f}{R_c}$. Hence, at $\alpha = 0$ there is no distortion of the shape and no difference between the image centroid and the projection of the three-dimensional centroid.

Equation (3.8) is unwieldly for analysis; however, it can be solved numerically. Figure 3.6 shows the image of the sphere found using this equation, for constant range from the lens centre and varying angles from the Z-axis. We can see that the projection of the sphere for angle $\alpha = 0$ is a circle, with the projection of the centroid in three dimensions matching the centroid of the two-dimensional image. As α increases, the eccentricity of the shape becomes more pronounced and the difference between the 3-D and 2-D centroids increases. Figure 3.7 shows the error between the image centroid and the 3-D centroid. This error has been normalized by the radius, thus giving indication of the distance from the true centre of the sphere to the estimated 3-D position of the sphere relative to the size of the sphere. We can see that at small angles from the longitudinal axis of the camera, there is little inaccuracy. As the angle increases, however, the error increases dramatically, and approaches infinity at $\alpha = 90°$. Therefore, with a lens that admits a narrow field of view, the error will not be large since α is

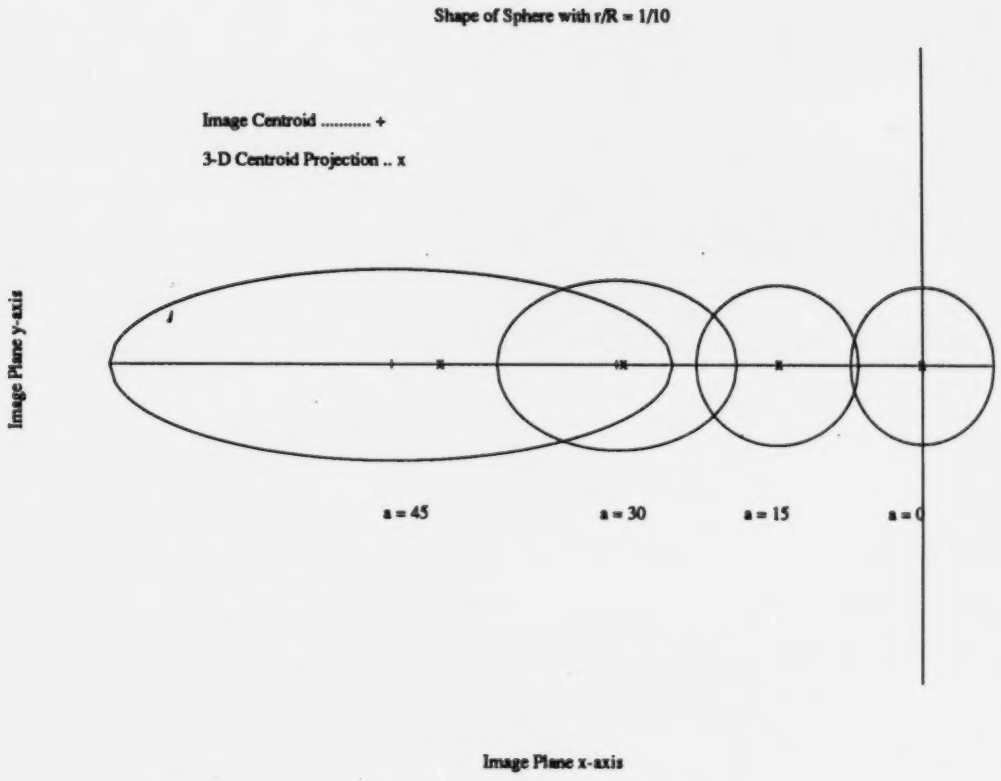


Figure 3.6: at $a(\alpha) = 0°, 15°, 30°$ and $45°$

constrained to be within a narrow range. For a lens with a wide field of view, however, the potential error will be much higher.

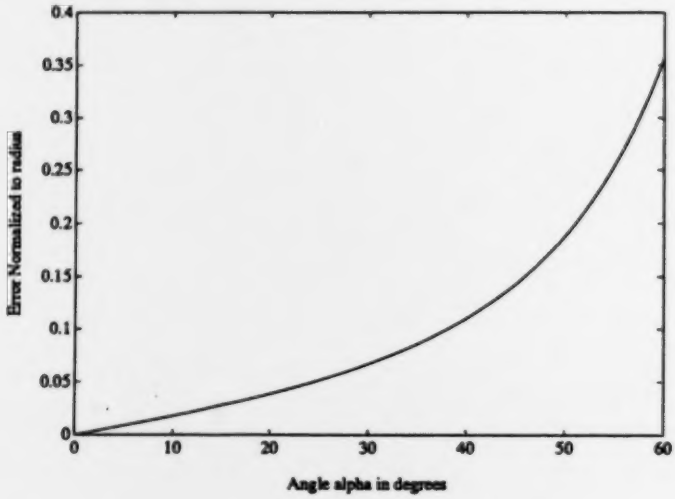


Figure 3.7: (R_c and r_c constant)

Chapter 4

Background Compensation

To be able to apply the motion detection techniques to be introduced in Chapter 5, we must compensate for the apparent motion of the background of a scene caused by camera motion. Our camera is mounted on a pan/tilt device and hence is constrained to rotate only. This is ideal for background compensation, since visual information is invariant to camera rotation [16].

Our objective in background compensation is to find a relationship between pixels representing the same 3-D position in images taken at different camera orientations. The projection of a 3-D point on the image plane is formed by a ray originating from the 3-D point and passing through the lens centre. The pixel representing this 3-D point is given by the intersection of this ray with the image plane (see Figure 3.1). If the camera rotates *about the lens centre*, this ray remains the same, since neither endpoints (the 3-D point and the lens centre) move due to this rotation.

Consequently, no previously viewable points will be occluded by other static points within the scene. This is important, since it implies that there is no fundamental change in information about a scene at different camera orientations. It should be noted that for theoretical considerations, the effect of the image boundary is ignored here. Obviously regions which pass outside of the image due to camera motion cannot be recovered. For camera rotation, the only components of our system that move are the camera coordinate system and the image plane. An example of this motion is shown in Figure 4.1.

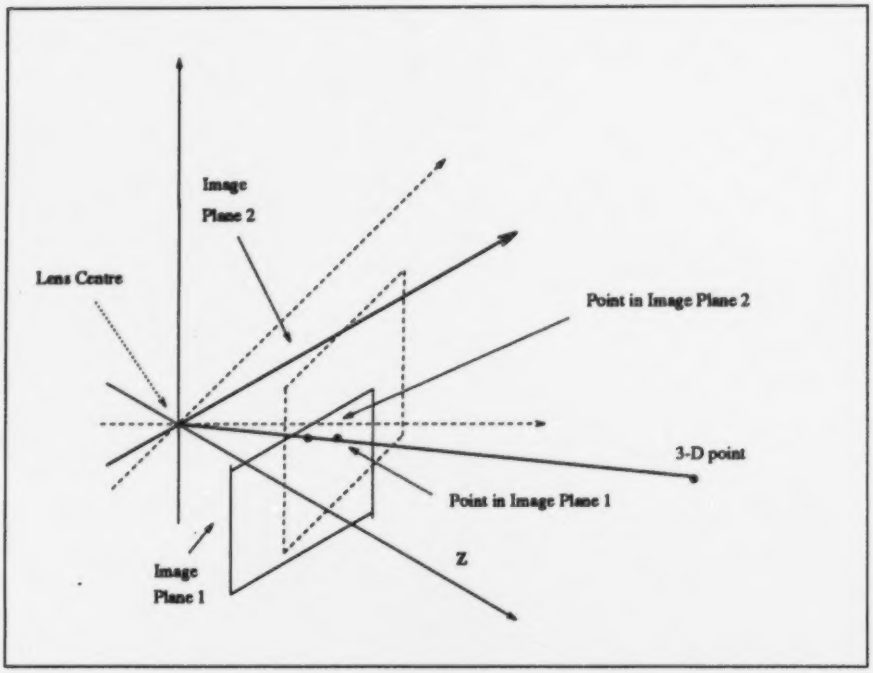


Figure 4.1: with the same lens centre. Notice the line between the 3-D point and the lens centre does not change with rotation of camera coordinate system.

For a given camera orientation, the reference frame we use is shown in Figure 3.3. The initial camera coordinate system can therefore be determined by a measure of the inclination of the camera due to previous rotation from the level position around the tilt axis. From this initial orientation, the camera undergoes small rotations about the pan and tilt axes, as shown in Figure 4.2.

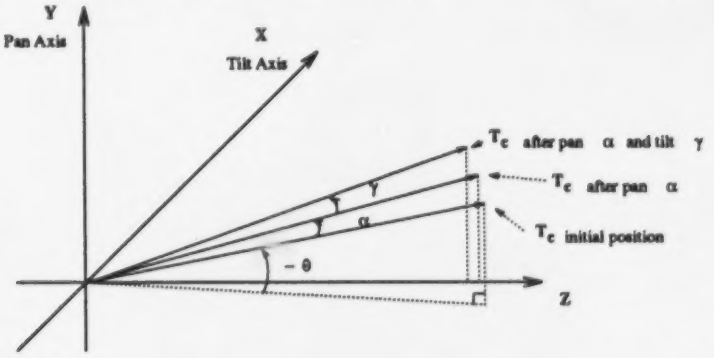


Figure 4.2: Camera transformations after pan and tilt rotations

We would like to obtain a relationship between every pixel position in the latest image with the corresponding pixel position in the previous image. For an initial inclination of the camera system θ, and pan and tilt rotations of α and γ respectively as showing in Figure 4.2, the relationship obtained is

$$x_{t-1} = f\frac{x_t + \alpha \sin\theta y_t + f\alpha\cos\theta}{-\alpha\cos\theta x_t + \gamma y_t + f} \tag{4.1}$$

$$y_{t-1} = f\frac{-\alpha\sin\theta x_t + y_t - f\gamma}{-\alpha\cos\theta x_t + \gamma y_t + f} \tag{4.2}$$

where f is the focal length.

With knowledge of f, θ, γ, and α, for every pixel position (x_t, y_t) in the current image we can calculate the position (x_{t-1}, y_{t-1}) of the corresponding pixel in the previous image.

4.1 Derivation of compensation algorithm

For each sampling instance, f, α, γ, and θ are known. If we provide the equation with all possible values of (x_t, y_t) for pixels within the image $I(t)$, we generate all

corresponding pixel locations in $I(t-1)$. The derivation of equations (4.1) and (4.2) follows.

Any point in the reference frame can be expressed as a vector P_r and is related to its position in the camera frame by the transformation

$$P_r = T_c P_c$$

where P_c is the point in the camera frame, and T_c is the 4×4 transform relating the camera frame to the reference frame.

Consider that the current camera frame, $[T_c(t)]$ is a result of a pan/tilt rotation from a previous camera position $[T_c(t-1)]$ as shown in Figure 4.2. Any point in the reference frame can be represented in camera frames before and after the motion as $P_c(t)$ or $P_c(t-1)$ and hence

$$P_r = T_c(t)P_c(t) = T_c(t-1)P_c(t-1)$$

It follows that the position of a point in one camera frame can be related to its position in the other camera frame by

$$P_c(t-1) = T_c(t-1)^{-1}T_c(t)P_c(t) \tag{4.3}$$

Since $T_c(t)$ is the result of applying pan and tilt rotations to $T_c(t-1)$

$$T_c(t) = Rot_Y(\alpha)Rot_X(\gamma)T_c(t-1) \tag{4.4}$$

$$T_c(t-1) = (Rot_Y(\alpha)Rot_X(\gamma))^{-1}T_c(t) \tag{4.5}$$

$$T_c(t-1)^{-1} = T_c(t)^{-1}Rot_Y(\alpha)Rot_X(\gamma) \tag{4.6}$$

By substituting equation (4.6) into equation (4.3) we obtain

$$P_c(t-1) = T_c(t)^{-1}Rot_Y(\alpha)Rot_X(\gamma)T_c(t)P_c(t) \tag{4.7}$$

where $Rot_Y(\alpha)Rot_X(\gamma)$ is simply

$$Rot_Y(\alpha)Rot_X(\gamma) = \begin{bmatrix} c\alpha & s\alpha s\gamma & s\alpha c\gamma & 0 \\ 0 & c\gamma & -s\gamma & 0 \\ -s\alpha & c\alpha s\gamma & c\alpha c\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.8}$$

However, since the sampling is assumed to be fast, the angles α and γ are assumed to be small ($< 5°$). For small angles, we can use the approximations

$$\left.\begin{array}{lcl} c\alpha, c\gamma & \approx & 1 \\ s\alpha, s\gamma & \approx & \alpha, \gamma \\ s\alpha s\gamma & \approx & 0 \end{array}\right\} \text{for } \alpha \text{ and } \gamma < 5°$$

Thus, the rotation matrix takes the form

$$Rot_Y(\alpha)Rot_X(\gamma) = \begin{bmatrix} 1 & 0 & \alpha & 0 \\ 0 & 1 & -\gamma & 0 \\ -\alpha & \gamma & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.9}$$

In general, a homogeneous transformation describing a frame takes the form

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \rho_x \\ r_{21} & r_{22} & r_{23} & \rho_y \\ r_{31} & r_{32} & r_{33} & \rho_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.10}$$

in which the 3x3 matrix $[r_{11}...r_{33}]$ represents the orientation of the frame and the vector $[\rho_x, \rho_y, \rho_z]$ represents a translation from the origin of the reference frame to the origin of the destination frame. In characterizing the frame transform, $T_c(t)$, we can make some simplifications to this general form to reduce the complexity of the derivation.

Since in our system there is no displacement in the X direction between the reference and camera frames, and there is no rotation about the Z-axis, the two X-axes will remain aligned. This reduces the transformation to

$$T_c(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & r_{22} & r_{23} & \rho_y \\ 0 & r_{32} & r_{33} & \rho_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.11}$$

This also makes the orientation matrix expressible via a single angle. The orientation can be considered to be a simple rotation about the X-axis (the current

tilt angle). Given an offset of angle θ, equation (4.11) becomes

$$T_c(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & -s\theta & \rho_y \\ 0 & s\theta & c\theta & \rho_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.12}$$

$$T_c(t)^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & s\theta & -c\theta\rho_y - s\theta\rho_z \\ 0 & -s\theta & c\theta & s\theta\rho_y - c\theta\rho_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.13}$$

Substituting in equations (4.9), (4.12), and (4.13), equation (4.7) can now be expanded to

$$P_c(t-1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & s\theta & -c\theta\rho_y - s\theta\rho_z \\ 0 & -s\theta & c\theta & s\theta\rho_y - c\theta\rho_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \alpha & 0 \\ 0 & 1 & -\gamma & 0 \\ -\alpha & \gamma & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & -s\theta & \rho_y \\ 0 & s\theta & c\theta & \rho_z \\ 0 & 0 & 0 & 1 \end{bmatrix} P_c(t) \tag{4.14}$$

After multiplication and simplification of terms using trigonometric identities, this becomes

$$P_c(t-1) = \begin{bmatrix} 1 & \alpha s\theta & \alpha c\theta & \alpha\rho_z \\ -\alpha s\theta & 1 & -\gamma & \gamma s\theta\rho_y - \gamma c\theta\rho_z \\ -\alpha c\theta & \gamma & 1 & \gamma c\theta\rho_y + \gamma s\theta\rho_z \\ 0 & 0 & 0 & 1 \end{bmatrix} P_c(t) \tag{4.15}$$

From this, by multiplying out these matrices we obtain

$$X_{t-1} = X_t + \alpha s\theta Y_t + \alpha c\theta Z_t + \alpha\rho_z \tag{4.16}$$

$$Y_{t-1} = -\alpha s\theta X_t + Y_t - \gamma Z_t + \gamma s\theta\rho_y - \gamma c\theta\rho_z \tag{4.17}$$

$$Z_{t-1} = -\alpha c\theta X_t + \gamma Y_t + Z_t + \gamma c\theta\rho_y - \gamma s\theta\rho_z \tag{4.18}$$

Dividing both sides of equation (4.16) by Z_{t-1} and multiplying by f we obtain

$$f\frac{X_{t-1}}{Z_{t-1}} = f\frac{X_t + \alpha s\theta Y_t + \alpha c\theta Z_t + \alpha\rho_z}{Z_{t-1}} \tag{4.19}$$

Now, recall from Section 3.2.2 that

$$x = \frac{fX}{Z} \qquad y = \frac{fY}{Z} \tag{4.20}$$

By applying equation (4.20) to the left hand side of equation (4.19) and substituting equation (4.18) for Z_{t-1} on the right hand side we obtain

$$x_{t-1} = f\frac{X_t + \alpha s\theta Y_t + \alpha c\theta Z_t + \alpha\rho_z}{-\alpha c\theta X_t + \gamma Y_t + Z_t + \gamma c\theta\rho_y - \gamma s\theta\rho_z} \tag{4.21}$$

Now dividing the top and bottom of the right hand side of equation (4.21) by Z_t and applying equations (4.20) again, this becomes

$$x_{t-1} = f\frac{x_t + \alpha s\theta y_t + f\alpha c\theta + \frac{f\alpha\rho_z}{Z_t}}{-\alpha c\theta x_t + \gamma y_t + f + f\frac{\gamma c\theta\rho_y + \gamma s\theta\rho_z}{Z_t}} \tag{4.22}$$

Notice that, aside from the last terms in the numerator and the denominator, this equation is now wholy dependent on image plane information. These two terms containing Z_t are present since the camera rotation is not being applied about the lens centre. If ρ_y and ρ_z were 0, this problem would not exist.

However, if we consider that f, ρ_y and ρ_z are significantly smaller than Z_t for actual implementation, the effect of this term will be small. Since it is virtually impossible for us to get any depth information about our scene, and the effect of these terms is small, we choose to neglect them with the understanding that the compensation achieved will not be perfect.

Hence, the final equation for x_{t-1} is

$$x_{t-1} = f\frac{x_t + \alpha\sin\theta y_t + f\alpha\cos\theta}{-\alpha\cos\theta x_t + \gamma y_t + f} \tag{4.23}$$

and similarly for y_{t-1}

$$y_{t-1} = f\frac{-\alpha\sin\theta x_t + y_t - f\gamma}{-\alpha\cos\theta x_t + \gamma y_t + f} \tag{4.24}$$

Chapter 5

Independent motion detection

5.1 Introduction

As we pointed out in Chapter 2, motion-energy detection is the most successful motion detection approach among practical, real-time tracking systems. Our implementation is therefore based primarily on motion-energy detection. Yet, because of the potential error incurred during camera motion compensation, modifications have to be made to the motion detection methods. In this chapter we will first discuss in more detail motion energy detection. Then we will describe what measures are taken in order to modify these techniques to active camera systems.

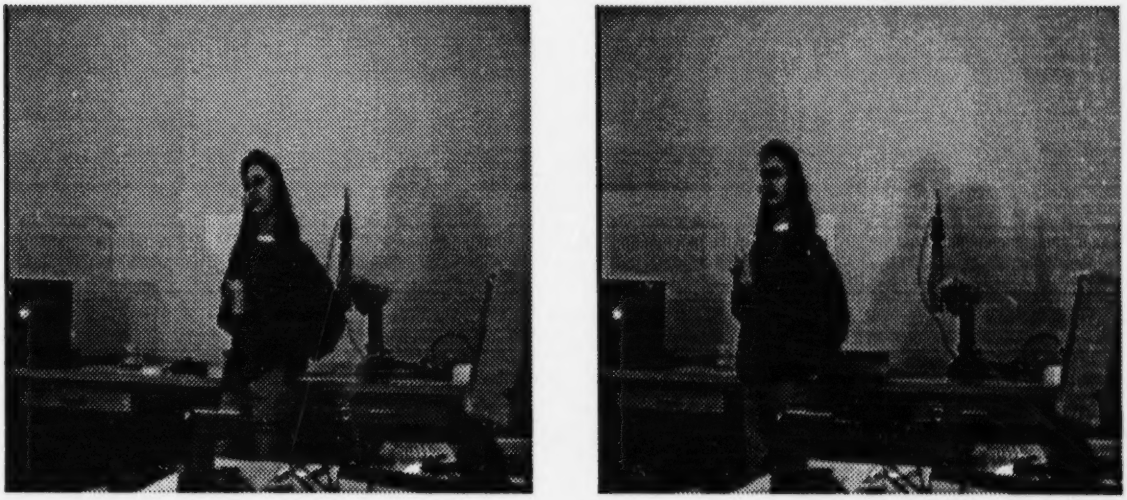
5.2 Motion detection with a static camera

In practice, motion-energy detection is implemented through spatio-temporal filtering. As the name implies, spatio-temporal filtering means filtering in both the spatial and temporal domains. The simplest implementation of motion energy detection is image subtraction. In this method, each image has the previous image in the image sequence subtracted from it, pixel-by-pixel. This is an approximation of the temporal derivative of the sequence. In equation form, the temporal

derivative is approximated by

$$\frac{dI(x,y,t)}{dt} \approx \frac{I(x,y,t) - I(x,y,t-\delta t)}{\delta t}$$

The absolute value of this approximation is taken and thresholded at a suitable level to segment the image into static and dynamic regions. Figure 5.1 shows two



Frame 1 Static Camera Sequence

Frame 2 Static Camera Sequence

Figure 5.1: Static camera sequence

images in an image sequence taken with a static camera. Figure 5.2 (left) shows the result of image subtraction with the centre of the area of motion marked by a cross.

As can be seen in Figure 5.2 (left), the drawback to this technique is that motion is detected in regions where the moving object was either at time t or $t - \delta t$. This means that the centre of the regions of motion will be close to the mid-point between the actual positions of the object at t and $t - \delta t$. For systems with a fast sampling rate (small δt) compared to the speed of the moving object, the difference in position of the object between frames will be small, and hence the midpoint between them may be adequate for rough position estimates. For objects with high speeds relative to the sampling rate, we must improve this

Figure 5.2: Result of thresholded image subtraction and edge detection

method. Our aim is to estimate the position of the moving object at time t. To achieve this, we use information available from the image taken at time t to extract the moving edges from the subtracted image shown in Figure 5.2 (left).

By applying edge detection filters, we can determine the edge strengths throughout the image. We obtain a binary edge image of the current frame by applying a threshold to the edge detection output. An example of the resultant edge image is shown in Figure 5.2 (right). One way to incorporate this information into the subtracted image is to perform a logical **AND** operation between the two binary images: the edge image and the subtracted image. This highlights the edges within the moving region to obtain the *moving edges* within the latest frame. Figure 5.3 (left) shows the result of this operation. As we can see, there will always be edges highlighted in the area previously occluded by the moving object. However, since these edges have only been viewed for one sample instant, it is unreasonable to expect the system to be able to detect whether or not they are moving until the next image is taken and processed.

A modified approach was suggested by Picton [22]. He argued that thresholds are empirically tuned parameters and to keep the system as simple and robust

Figure 5.3: Moving edges detected in frame 2 of static camera sequence

as possible, the number of tuned parameters should be minimized. Hence, he proposed reducing thresholding to a single step by multiplying the pre-thresholded values of the edge strength and image subtraction to obtain a value indicative of both edge strength and temporal change combined. This product is then thresholded and thus the tuned parameters are reduced to a single threshold. The result of this multiplication method is shown in Figure 5.3 (right). As we can see, the boundary of the moving object is the same as with the logical **AND** method. However, since more interior edges are also emphasized, the centroid of the moving edges is closer to the true centre of the moving object. Figure 5.4 shows the steps taken in implementing Picton's method of motion detection with a static camera.

5.3 Motion detection by an active camera

For a stationary camera, the pixel-by-pixel subtraction described in Section 5.2 is possible, since with a static scene a given 3-D point will continuously project to the same position in the image plane. For a moving camera this is not the

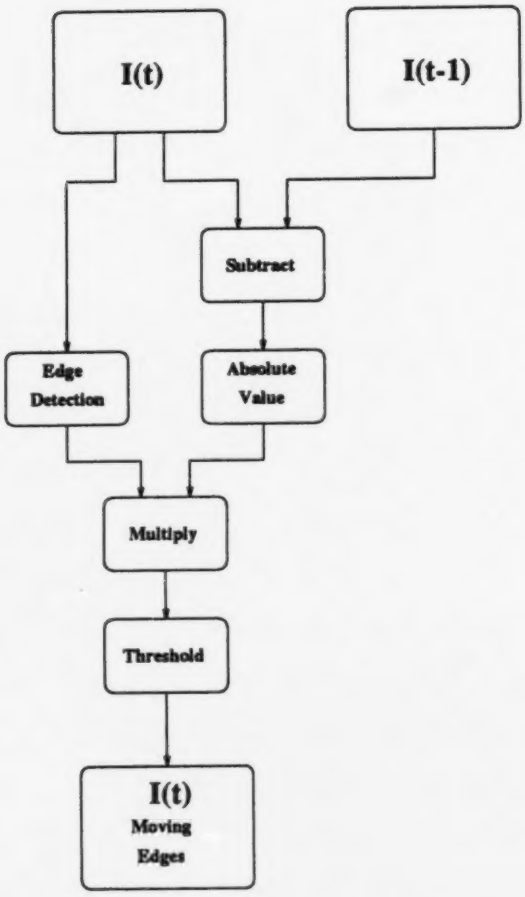


Figure 5.4: Motion detection with a static camera

case. To apply pixel-by-pixel comparison with an active camera image sequence, we must map pixels which correspond to the same 3-D point to the same image plane position.

Chapter 4 has outlined the geometry behind invariance to rotation and derives the mapping function between images. For each pair of images processed in the image sequence, the image at time $t - \delta t$ is mapped so as to correspond pixel-by-pixel with the image at time t. Regions with no match between the two images are ignored. The image subtraction, edge extraction and subsequent moving edge detection is done as detailed in Section 5.2. The active camera motion detection method is summarized in the block diagram in Figure 5.5.

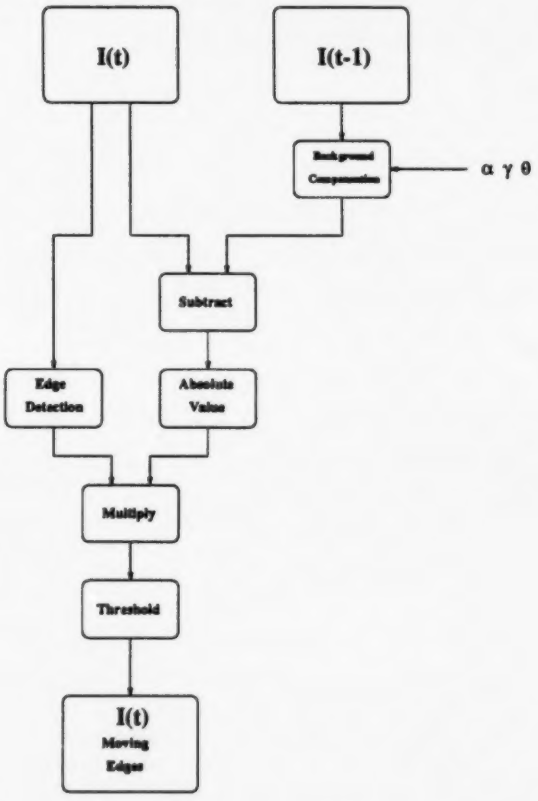
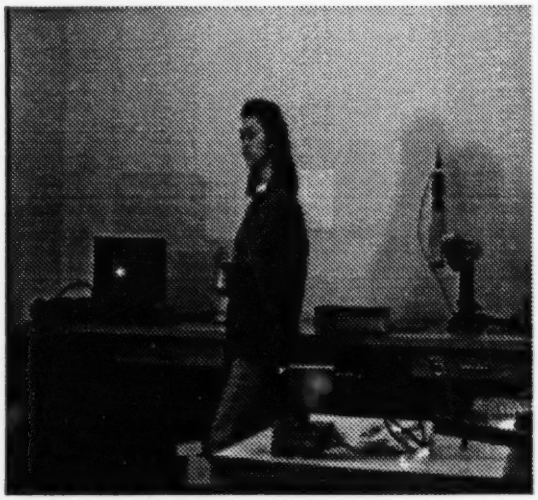


Figure 5.5: Motion detection with an active camera

Frame 1 Moving camera sequence



Frame 2 Moving camera sequence

Figure 5.6: Moving camera sequence

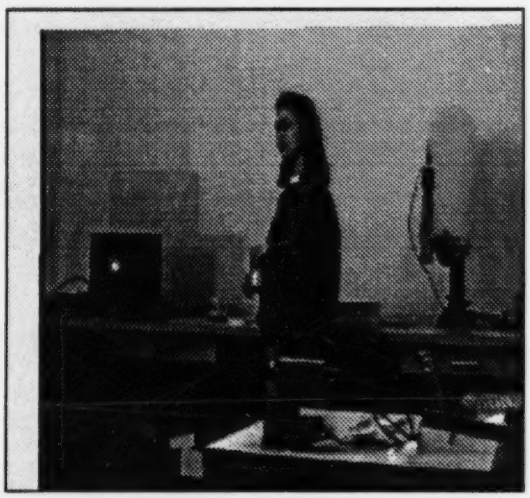
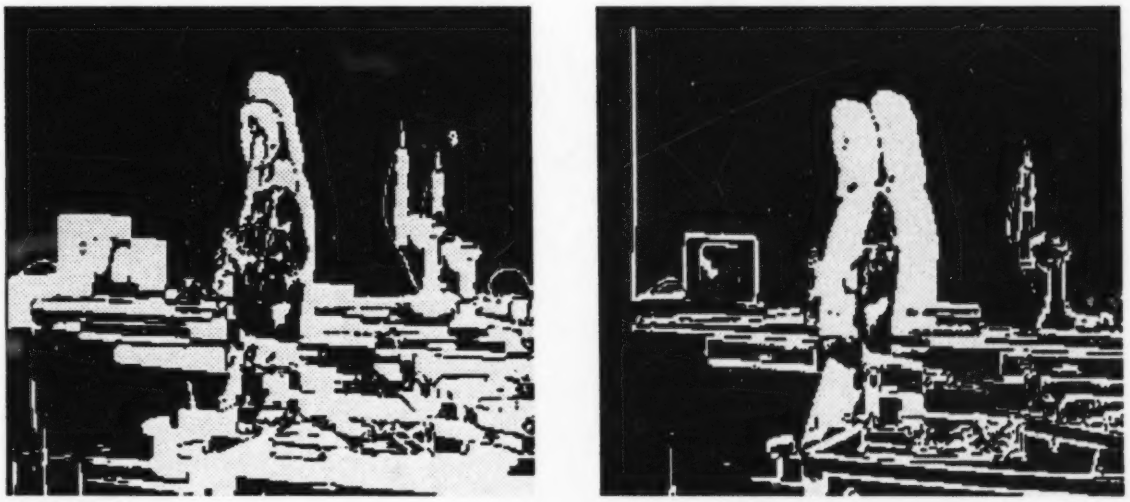


Figure 5.7: Image 1 of the moving camera sequence compensated for camera motion



Subtraction of images without compensation for camera rotation

Subtraction of images with motion compensation (note inaccuracies due to compensation error)

Figure 5.8: Image subtraction with and without compensation

Figure 5.6 shows two images taken from different camera orientations. Figure 5.7 contains the first image in the moving camera sequence with the compensation mapping applied. Notice the white border along the top and left sides of the image. These regions have no overlap between the two images and hence are ignored. Figure 5.8 (left) shows the results of image subtraction without compensation. Clearly this is unsuitable. The object of interest appears to be moving less than the background. Figure 5.8 (right) shows the results after background compensation. Notice that the scene background components have *not* been entirely eliminated. This is due to inaccuracies in the inputs to the compensation algorithm, and approximations made in the algorithm derivation.

The background compensation algorithm presented in Chapter 4 was derived with the assumption that rotation occurs about the lens centre. In reality this is not the case for our system, and the small amount of camera translation corrupts the compensation method. As well, errors in pan/tilt position sensors and camera calibration will contribute to the compensation inaccuracy. The following section will show how we overcome this compensation noise and improve the robustness of our method.

5.4 Robust motion detection with an active camera

If we could achieve exact background compensation, the methods described so far would be sufficient. In the presence of position inaccuracies, however, the results of these methods rapidly deteriorate. We are using edge information in our techniques to detect moving objects. Ironically, regions with good edge characteristics are the most sensitive to compensation errors during image subtraction. That is to say, false motion caused by inaccurate compensation will be greatest in strong edge regions, yet these are the very regions that are considered as candidates for moving edge pixels. This makes the previously presented method unreliable.

Since errors in angle information are inevitably present, it is desirable to develop methods of motion detection that can robustly reject the false motion they cause. Errors in pan/tilt angles can be due to sensor error. For a real-time system with a continuously moving camera, there is the additional problem of synchro-

nization. If the instances of grabbing an image and reading position sensors are not perfectly synchronized, the finite difference in time between these events can be considered as error in position sensing. This error is calculated as

$$\theta_e = \omega \times \Delta t$$

where θ_e is the error in angular position, ω is the angular velocity of rotation and Δt is the synchronization error. Since few vision systems are designed with this consideration in mind, the problem is a common one in active vision applications.

Figure 5.8 (right) shows an example of the results of image subtraction after inaccurate background compensation. Notice the region of the moving object contains a broad area where true motion was detected, whereas false motion is characteristically narrow bands bordering the strong edges of the scene background. Our approach to removing the false motion utilizes the expectation of a wide region of true motion being present. By using morphological erosion and dilation (morphological *opening*) we eliminate narrow regions of detected motion, while preserving the original size and shape of the wide regions.

5.4.1 Morphological Filtering

Morphological filters applied to digital images have been used for several applications which include: edge detection, noise suppression, region filling and skeletonizing [20]. Morphological filtering is essentially an application of set theory to digital signals. It is implemented with a mask **M** overlaying an image region **I** as shown in Figure 5.9.

For morphological filtering, the image pixel values, namely

$$\begin{bmatrix} i_{11} & \cdots & i_{1n} \\ \vdots & \ddots & \vdots \\ i_{n1} & \cdots & i_{nn} \end{bmatrix}$$

are selected by the values of **M** as members of a set for analysis with set theory methods. Usually, values of elements in a morphological filter mask are either 0 or 1. If the value of the filter mask element is 0, the corresponding pixel value is not a member of the set. If the value is 1, the pixel value is included in the set.

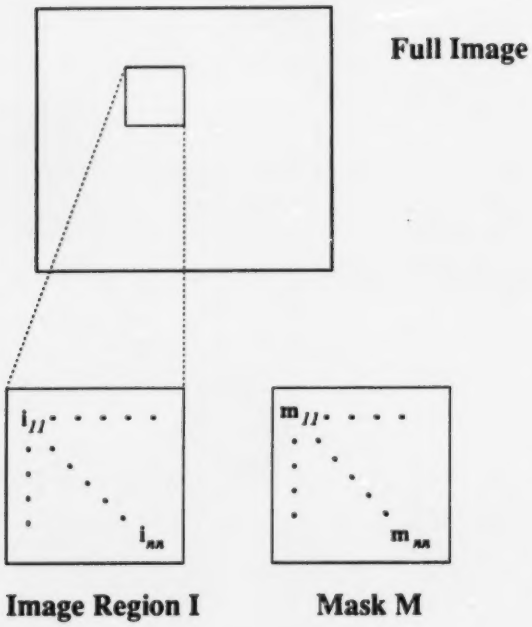


Figure 5.9: Example of mask for morphological filters

We can express the elements of the image region selected by the morphological mask as a set A such that:

$$A = \{i_{jk} | 1 \leq j \leq n, 1 \leq k \leq n, m_{jk} = 1\}$$

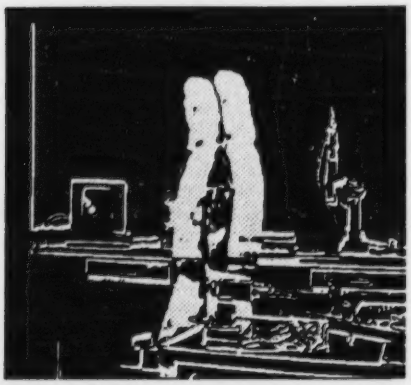
The morphological operations we will consider are *erosion* and *dilation*. Erosion of A is: $E_A = min(A)$
Dilation of A is: $D_A = max(A)$

For binary images, this equates to:
$E_A = 0$ if *any* element of $A = 0$
$E_A = 1$ if *all* elements of $A = 1$
$D_A = 1$ if *any* element of $A = 1$
$D_A = 0$ is *all* elements of $A = 0$

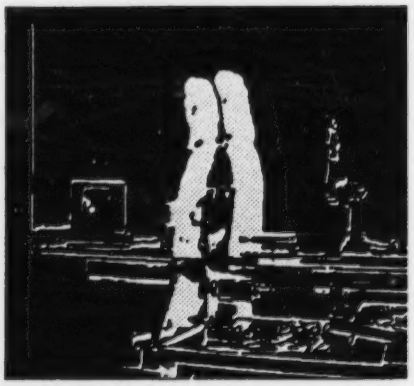
For a 3×3 morphological mask with all elements set to 1, and a binary image, this is also known as *shrinking* and *growing*.

By applying erosion to the subtracted image, narrow regions can be eliminated. If the regions to be preserved are wider than the filter mask, they will only be thinned and not completely eliminated. After dilation by a mask of the same size, they will be roughly restored to their original shape and size. If the erosion mask is wider than a given peak region, that region will be eliminated completely and not appear after dilation.

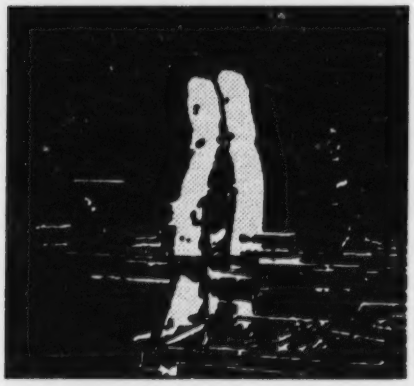
Figure 5.10 shows the subtracted image of Figure 5.8 (bottom) eroded by different size masks. For this particular image sequence we can see that to completely eliminate the noise due to position inaccuracies, we must use a mask size of 9×9 or 11×11.



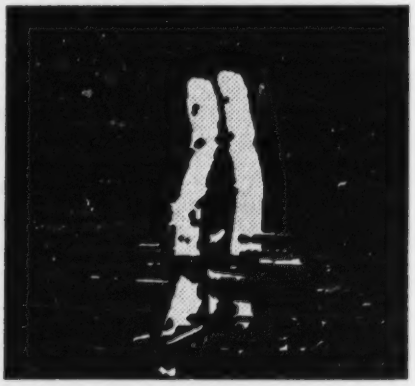
After compensation and image subtraction



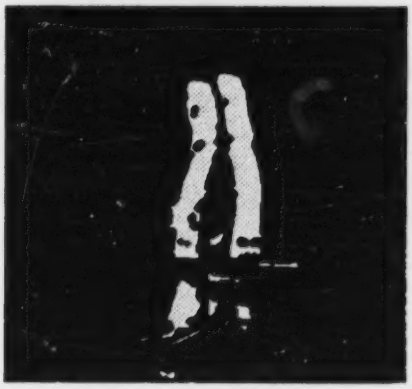
After erosion with a 3x3 mask



After erosion with a 5x5 mask



After erosion with a 7x7 mask



After erosion with a 9x9 mask

After erosion with an 11x11 mask

Figure 5.10: Subtracted Image with various sizes of erosion masks applied

Chapter 6

Experimental Results

6.1 Overview

The experimental results presented here are from two sequences of images taken with the pan/tilt mounted camera. The first sequence uses a camera aligned horizontally (i.e. no tilt) and restricts the camera motion to pan. The second sequence is taken with both pan and tilt camera motion. The image sequences are processed off-line, and hence the pan/tilt motion of the camera is not controlled by the motion detection results.

Two methods of motion detection were tested: thresholding the temporal and spatial derivatives independently, and multiplication of derivatives prior to thresholding. The image sequences and the processed results are presented in this chapter.

6.2 Equipment layout

This section describes the system used for experimentation. The camera is mounted on the Cohu-MPC, a pan/tilt device. Instruction for this device are sent from a SUN3 over a serial interface. The SUN3 is mounted on a VME-bus with the DT1451 frame digitizer board. The camera is a CCD device with standard video output.

The Cohu-MPC allows controls of rotation about two axes (pan and tilt) as

well as adjustment of zoom and focus settings. The position sensing of the pan/tilt axes is done by a potentiometer coupled to the driving motor shaft.

6.3 Experimental results

Figures 6.1 and 6.6 show two image sequences taken with the pan/tilt mounted camera. Figure 6.1 (image sequence 1) was taken with camera motion constrained to pan only, whereas Figure 6.6 (image sequence 2) had pan and tilt motion. Figures 6.2, 6.4, 6.7 and 6.9 show the results of two motion detection methods applied to these image sequences with the moving edges only visible. Figures 6.3, 6.5, 6.8 and 6.10 show the results of the motion detection overlaid upon the original images so that sources of the edges are more readily apparent.

6.4 Discussion of results

The results shown have been generated using the two motion detection techniques presented in Chapter 5. The two approaches are summarized here:

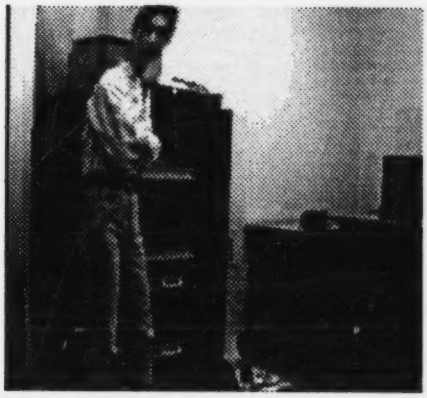
Approach 1 Binary images of the spatial and temporal derivative peaks are formed by thresholding the subtracted and edge strength images. These two binary images are then **AND**ed together to extract the moving edges in the scene.

Approach 2 The unthresholded values of the spatial and temporal derivatives are multiplied, and the product is thresholded to extract the moving edges.

Both approaches use two 3×3 sobel edge detection kernels to find the edge strength in the vertical and horizontal directions. The edge strength used for motion detection is simply found by

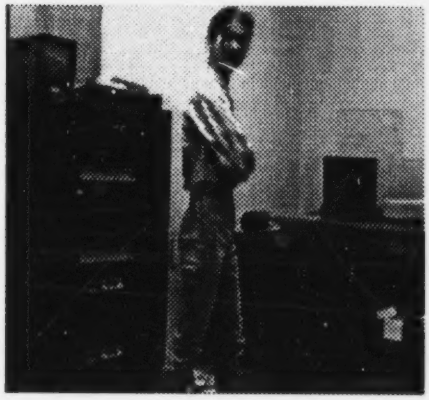
$$STRENGTH = \sqrt{(df/dx)^2 + (df/dy)^2}$$

The results of the two approaches on image sequence 1 are shown in Figures 6.2 and 6.4. Although both approaches work, they exhibit significantly different characteristics. Approach 1 detects primarily the boundary of the moving object,

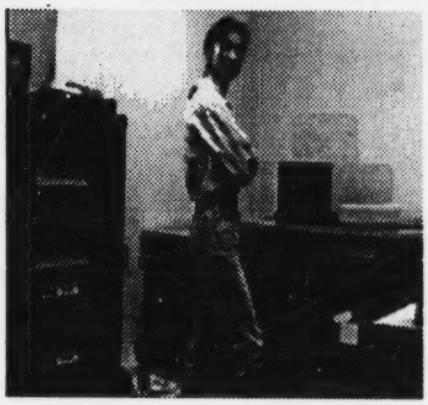


Frame 1

Frame 2

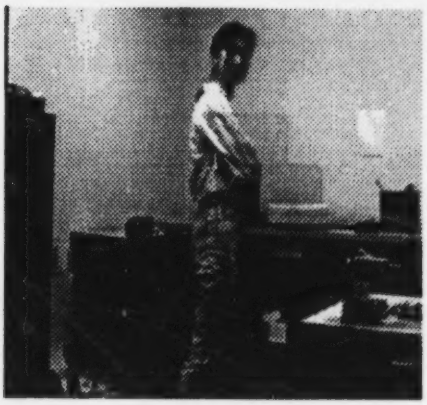


Frame 3



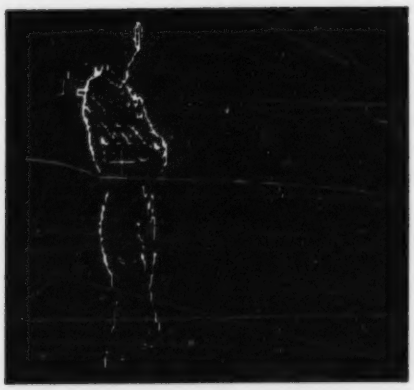
Frame 4

Frame 5

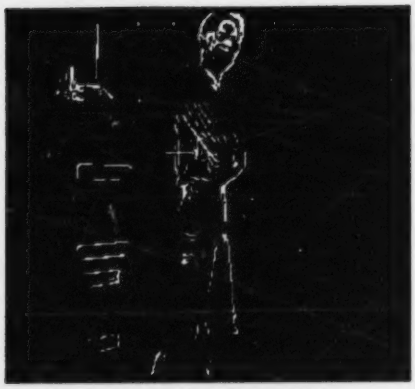


Frame 6

Figure 6.1: Pan-only image sequence



Frame 2 moving edges



Frame 3 moving edges

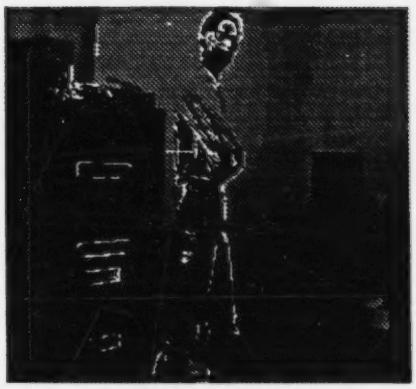
Frame 4 moving edges

Frame 5 moving edges

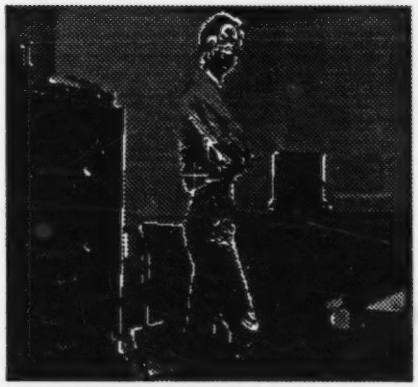
Frame 6 moving edges

Figure 6.2: Moving edges in pan-only image sequence (Approach 1)

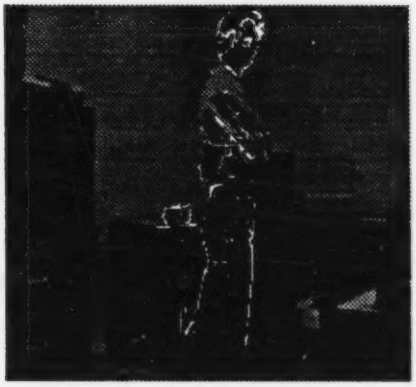
Frame 2 moving edges



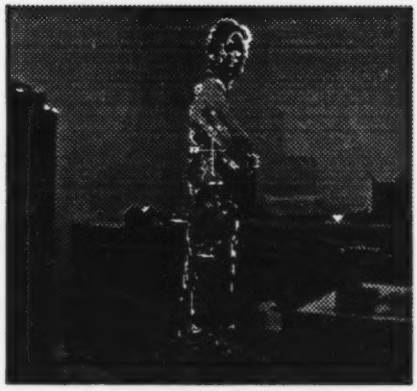
Frame 3 moving edges



Frame 4 moving edges

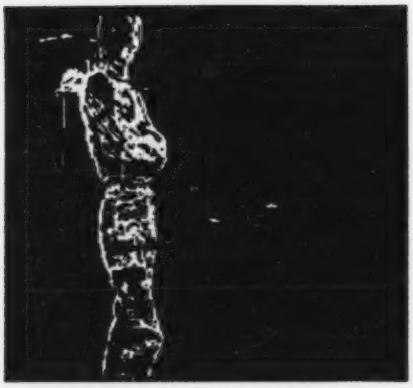


Frame 5 moving edges



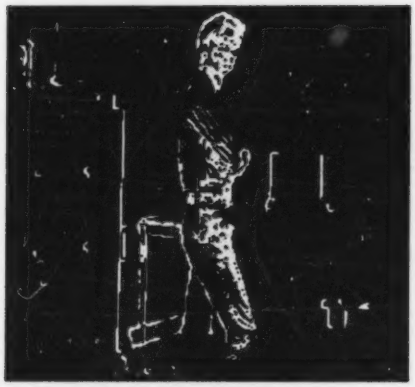
Frame 6 moving edges

Figure 6.3: Moving edges in pan-only image sequence overlaid upon the originals (Approach 1)



Frame 2 moving edges

Frame 3 moving edges

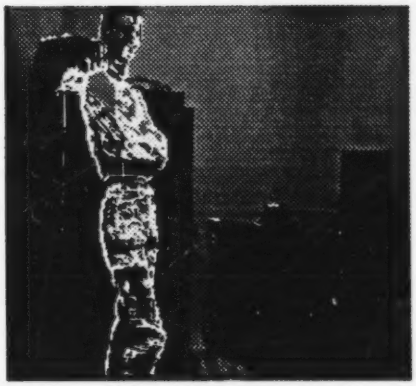


Frame 4 moving edges

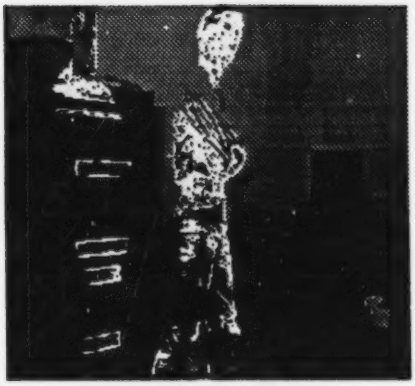
Frame 5 moving edges

Frame 6 moving edges

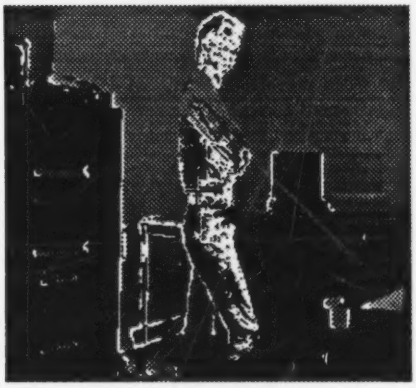
Figure 6.4: Moving edges in pan-only image sequence (Approach 2)



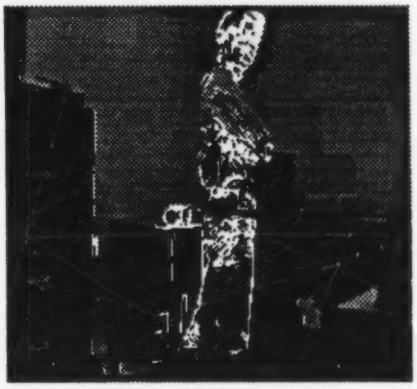
Frame 2 moving edges



Frame 3 moving edges



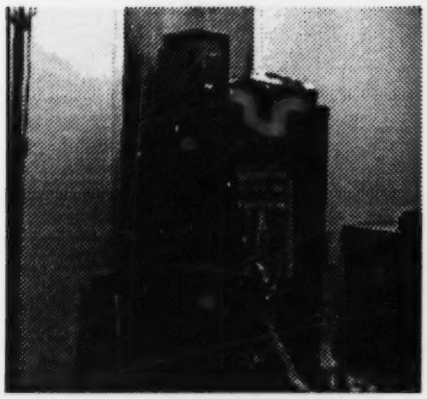
Frame 4 moving edges



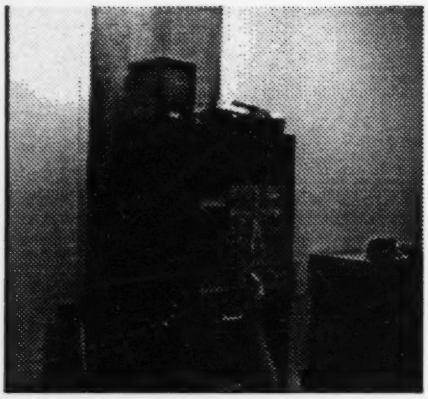
Frame 5 moving edges

Frame 6 moving edges

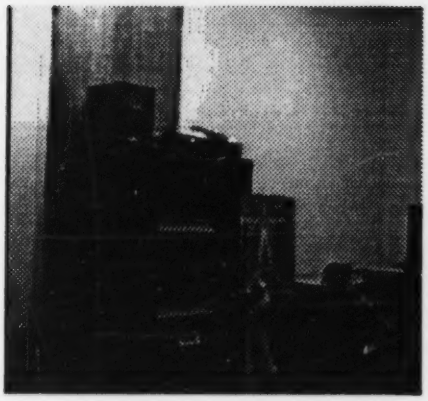
Figure 6.5: Moving edges in pan-only image sequence overlaid upon the originals (Approach 2)



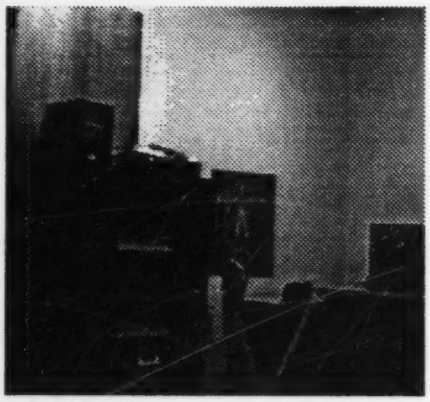
Frame 1



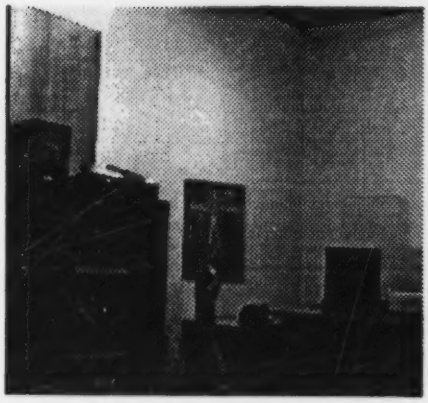
Frame 2



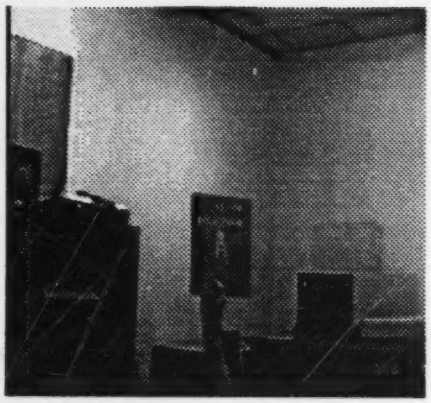
Frame 3



Frame 4

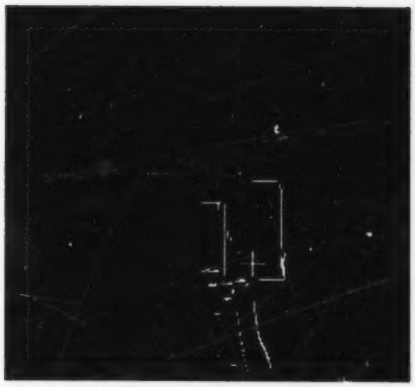


Frame 5

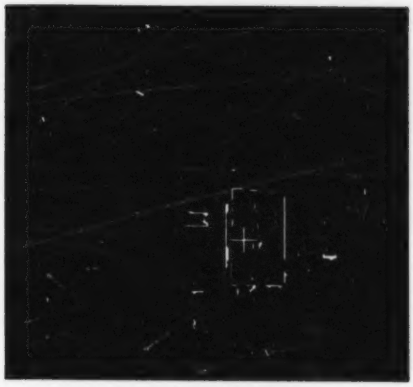


Frame 6

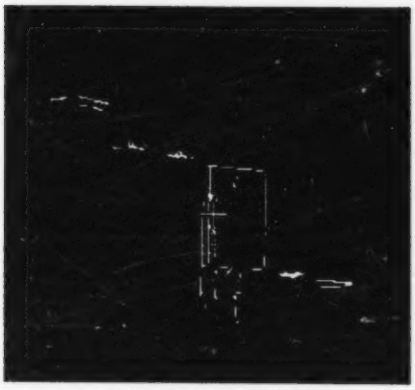
Figure 6.6: Pan and tilt image sequence



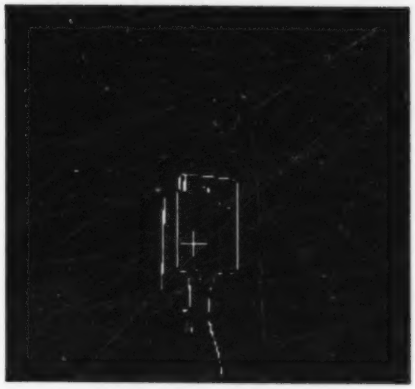
Frame 2 moving edges



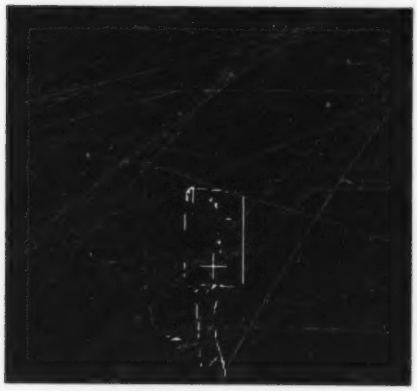
Frame 3 moving edges



Frame 4 moving edges

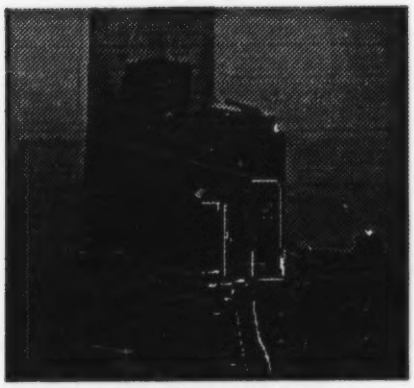


Frame 5 moving edges

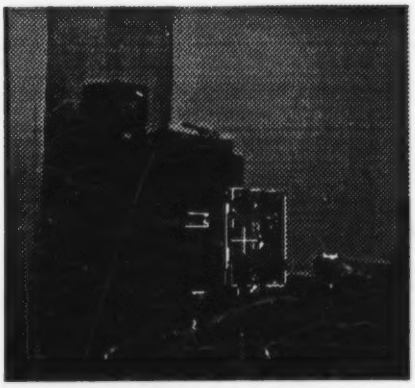


Frame 6 moving edges

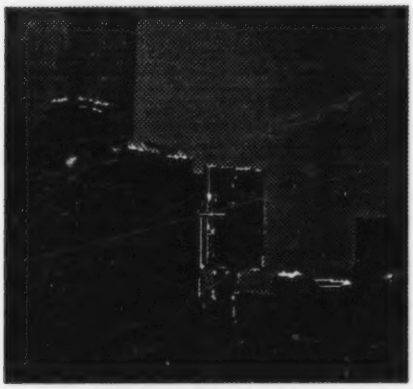
Figure 6.7: Moving edges in pan and tilt image sequence (Approach 1)



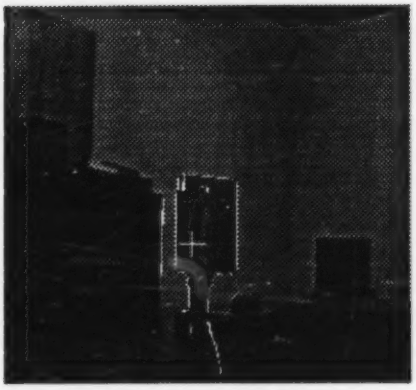
Frame 2 moving edges



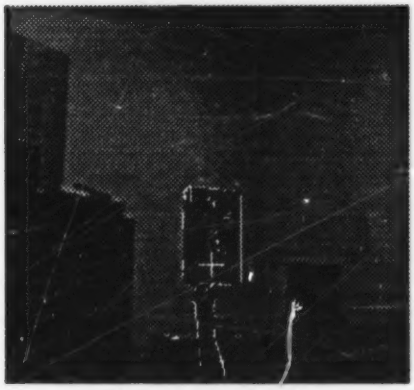
Frame 3 moving edges



Frame 4 moving edges

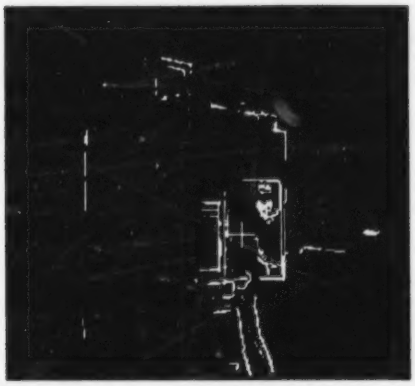


Frame 5 moving edges

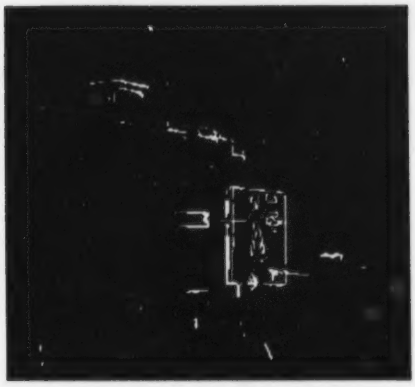


Frame 6 moving edges

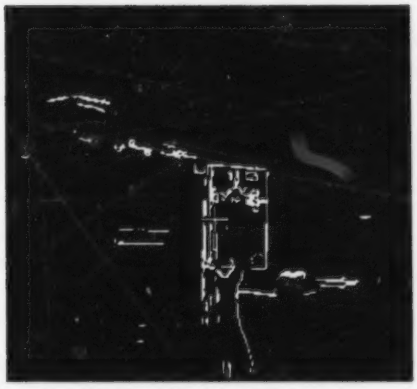
Figure 6.8: Moving edges in pan and tilt image sequence overlaid upon the originals (Approach 1)



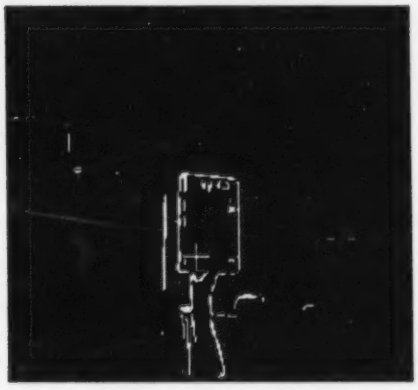
Frame 2 moving edges



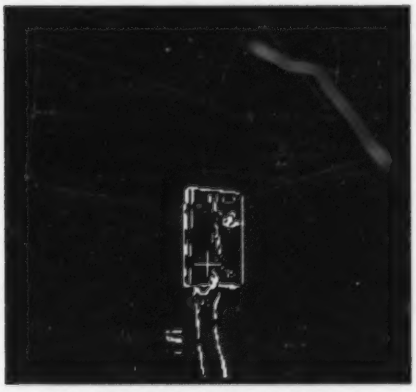
Frame 3 moving edges



Frame 4 moving edges

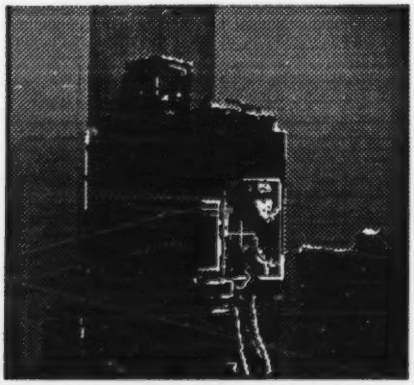


Frame 5 moving edges

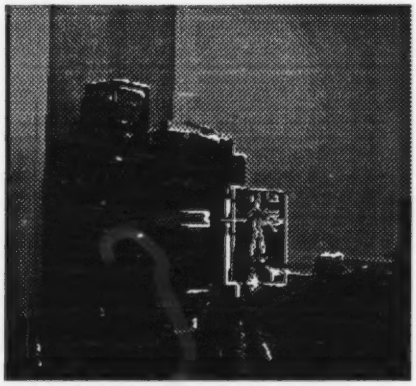


Frame 6 moving edges

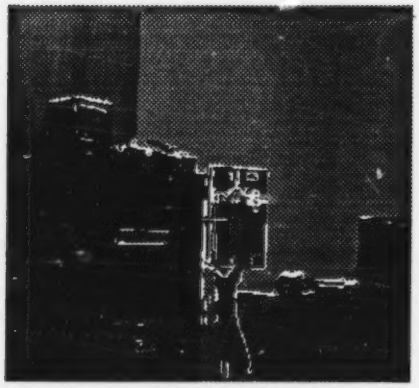
Figure 6.9: Moving edges in pan and tilt image sequence (Approach 2)



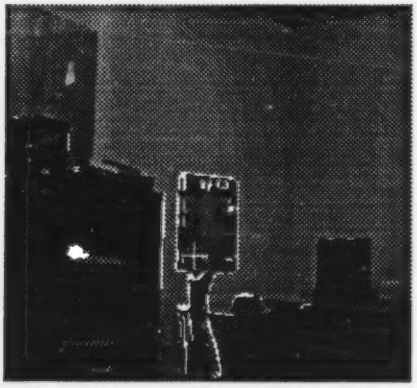
Frame 2 moving edges



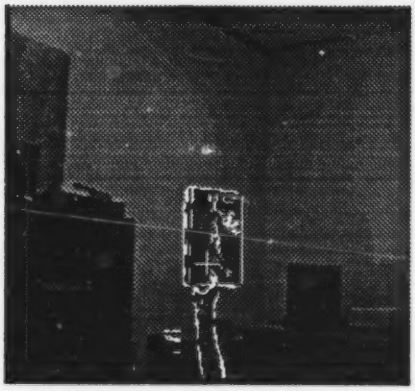
Frame 3 moving edges



Frame 4 moving edges



Frame 5 moving edges



Frame 6 moving edges

Figure 6.10: Moving edges in pan and tilt image sequence overlaid upon the originals (Approach 2)

since the independent thresholding of the edge image tends to eliminate edges within the image of the object. Approach 2 does not remove the contribution of the fainter edges until after multiplication with the temporal derivative. Hence we find that many interior edges of the moving object are revealed.

From the results shown in Figure 6.4, Approach 2 seems preferable, since it produces a stronger signal within the moving object and thus provides a more robust centroid of motion. From Figure 6.9, however, we see that for a different moving object, this advantage is lost. In the case of image sequence 2, the moving object is a book which does not have much texture on its surface. Therefore, due to the lack of internal edges, the benefits of Approach 2 are not as significant as in image sequence 1. It should also be mentioned that Approach 1 is easier and faster to implement on the hardware available.

In image sequence 1, the moving object is a person. Due to the folds and creases in the clothing, the moving region has rich texture. This provides a dense area of weak edges that can be brought out by Approach 2. The backgrounds of the image sequences, on the other hand, are characterized by homogeneous blocks of similar intensity, bordered by abrupt changes in greyscale. In the results of both approaches, background edges which were occluded by the moving object in the previous frame are detected as moving edges. Since our system has only viewed these regions for a single frame it is unreasonable to expect the algorithm to determine whether these edges are static or dynamic until the next frame is processed. Yet, if the background had more varied texture, such as a wheat field or a chain-link fence, regions previously blocked by the moving object would have the same weak edges brought out by Approach 2. This would corrupt the moving edge signal and tend to produce a centroid of the moving object which would lag the object's true position.

6.5 Inaccuracies in moving edge detection

In Figures 6.2, 6.4, 6.7 and 6.9 the detected moving edges of the two images sequences are shown. It is evident that some spurious motion has been detected. As well, certain regions which are moving have not been detected.

To begin, we will discuss the moving edges that are missing. These are primarily in the first image sequence along the moving person's shirt. We can see from the image that there is very low contrast between the shirt and the light wall behind it. This is an inherent difficulty with image subtraction. Since images are noisy signals, small changes in greyscale must be discounted for robustness considerations. Hence, without good contrast, the image subtraction will fail. Possible solutions to this are optimal thresholding and use of color images, which are briefly discussed in Section 8.3.

The false motion detected in both image sequences are due to either previous occlusion (as discussed above) or inaccurate background compensation with insufficiently large morphological filter mask. In image sequence 1, for example, we can see that the moving edges detected by both techniques in frame 4 have significant false motion present. Specifically, motion is detected along the dresser border and the TV monitor seen in the background. This is due to particularly poor position readings for the frame 3-4 pair. This false motion can be eliminated by increased filtering. An increase in filter size places additional computational burden on the filtering stage, as well as possibly eliminating the true motion signal. The relationship between position noise and filtering requirements is presented in Chapter 7.

Chapter 7

Analysis of compensation inaccuracy

7.1 Introduction

As discussed in Chapter 5, noisy position information corrupts background compensation algorithm and necessitates additional noise removal techniques. Morphological filtering has been presented as one technique to remove narrow regions of false motion from subtracted images. For effective noise removal to occur, the morphological erosion mask must be at least as wide as the regions of false motion. If the mask is not wide enough, some noise will remain after erosion and will be expanded to its original size during dilation. This means that no noise will be removed.

This method of noise removal is therefore an *all-or-nothing* approach. The advantage is, for acceptable noise levels, false motion is completely removed. The disadvantage is, if the noise exceeds the filter capacity, no noise removal takes place. Because of this behavior, it is important that we use filters large enough to completely remove the expected noise. However, for computational reasons, it is also desirable to limit filtering to the minimum required. This motivates us to investigate the relationship of noise characteristics to filtering requirements.

Recall the mapping algorithm derived in Chapter 4,

$$x_{t-1} = f\frac{x_t + \alpha \sin\theta y_t + f\alpha\cos\theta}{-\alpha\cos\theta x_t + \gamma y_t + f} \tag{7.1}$$

$$y_{t-1} = f\frac{-\alpha\sin\theta x_t + y_t - f\gamma}{-\alpha\cos\theta x_t + \gamma y_t + f} \tag{7.2}$$

The error between the correct pixel position and the pixel position found with inaccurate angle information in the mapping algorithm can be expressed as

$$x_e = x_{t-1}(\alpha,\gamma) - x_{t-1}(\alpha+\Delta_\alpha,\gamma+\Delta_\gamma) \tag{7.3}$$

$$y_e = y_{t-1}(\alpha,\gamma) - y_{t-1}(\alpha+\Delta_\alpha,\gamma+\Delta_\gamma) \tag{7.4}$$

Where x_e and y_e are the errors in mapped pixel position in the x and y directions, and Δ_α and Δ_γ are inaccuracies in measurement of the rotations α and γ respectively.

For evaluating the error in pixel mapping, we consider several cases depending on the location of (x_t, y_t). In general, the error in the mapped pixel position is greater as we move further from the centre of the image. We use pixel positions in the image centre to simplify the error equations when determining general error characteristics as well as border pixels to determine the worst case behavior. To simplify our discussion θ is constrained to 0, i.e. the camera at the level position.

7.2 Compensation error for pan-only rotation

For the pan-only case, γ, the tilt rotation, is 0. Hence equations (7.1) and (7.2) is reduced to

$$x_{t-1} = f\frac{x_t + f\alpha}{f - \alpha x_t} \tag{7.5}$$

$$y_{t-1} = f\frac{y_t}{f - \alpha x_t} \tag{7.6}$$

To evaluate $x_{t-1}(\alpha+\Delta_\alpha), y_{t-1}(\alpha+\Delta_\alpha)$, we will approximate the function with a first order Taylor series expansion as follows:

$$x_{t-1}(\alpha+\Delta_\alpha) = x_{t-1}(\alpha) + \frac{\partial x_{t-1}}{\partial\alpha}\Delta_\alpha \tag{7.7}$$

$$y_{t-1}(\alpha + \Delta_\alpha) = y_{t-1}(\alpha) + \frac{\partial y_{t-1}}{\partial \alpha}\Delta_\alpha \tag{7.8}$$

Substituting equations (7.7) and (7.8) into our equations for error in the compensated pixel position [equations (7.3) and (7.4)] we obtain

$$x_e = \frac{\partial x_{t-1}}{\partial \alpha}\Delta_\alpha = f\frac{x_t^2 + f^2}{(f - \alpha x_t)^2}\Delta_\alpha \tag{7.9}$$

$$y_e = \frac{\partial y_{t-1}}{\partial \alpha}\Delta_\alpha = f\frac{y_t x_t}{(f - \alpha x_t)^2}\Delta_\alpha \tag{7.10}$$

The magnitude of the error in pixel position, e, is shown in Figure 7.1 and can be expressed as

$$e^2 = x_e^2 + y_e^2 \tag{7.11}$$

For pan-only rotation, the error is predominantly in the x direction, since the

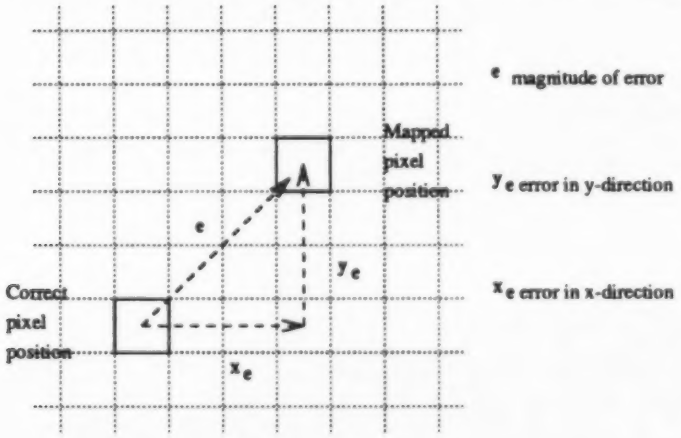


Figure 7.1: Magnitude of pixel mapping error

change in the y component for pixels at different viewpoints is effected by changes in perspective only. Therefore

$$e^2 \approx x_e^2 \qquad e \approx x_e$$

To determine the pan-angle error Δ_α, for a given pixel mapping error, from equation (7.9) we obtain

$$\Delta_\alpha = \frac{x_e(f - \alpha x_t)^2}{f(x_t^2 + f^2)} \tag{7.12}$$

x_e in pixels	$\Delta\alpha$ in degrees	y_e in pixels	e in pixels
1	0.056555	0.075864	1.002873
2	0.113111	0.151728	2.005747
3	0.169666	0.227592	3.008620
4	0.226222	0.303456	4.011429
5	0.282777	0.379320	5.028694
6	0.339333	0.455184	6.017241
7	0.395888	0.531049	7.020114
8	0.452444	0.606913	8.022988
9	0.508999	0.682777	9.025862
10	0.565555	0.758641	10.028736
11	0.622110	0.834505	11.031609

Table 7.1: Pan-only compensation error

Once Δ_α is determined, we can solve for y_e using equation (7.10) to verify our initial assumpt that y_e is negligible. For our system, where $f = 890$ and the maximum $x_t = 255$ we can make a table of values of Δ_α for given errors x_e, the corresponding y error for this position, y_e and e for magnitude of (x_e, y_e). The value for α used was 5°, since this is the upper bound for which our system is designed. The results are shown in Table 7.1; notice the relationship between Δ_α and x_e is linear.

7.3 Compensation error for pan and tilt rotations

The error in pixel mapping is more difficult to obtain if both pan and tilt rotations are made. However, to gain insight in the general characteristics of the error, we will consider a special case, where $(x, y) = (0, 0)$, which is the pixel that lies directly along the Z-axis of the camera coordinate system.

For this case, the pixel mapping functions are

$$x_{t-1} = f\alpha \tag{7.13}$$

$$y_{t-1} = f\gamma \tag{7.14}$$

and, our error equations become

$$x_e = f\Delta_\alpha \tag{7.15}$$

$$y_e = f\Delta_\gamma \tag{7.16}$$

From equation (7.11) we know the magnitude of the error can be expressed by

$$e^2 = x_e^2 + y_e^2 = f^2{\Delta_\alpha}^2 + f^2{\Delta_\gamma}^2 \tag{7.17}$$

Plotting lines of constant error in terms of Δ_α and Δ_γ yields a series of concentric circles with radii of

$$r = \frac{e}{f}$$

for any constant error e.

Unfortunately, the pixel error at the pixel centre is not the worst case. The circles described by equation (7.17) are for error at the image centre. Although this characterizes the compensation-error versus angle-error, it does not show the worst case we can expect. To estimate the worst-case error, we use the worst-case x_e for no tilt error and the worst-case y_e for no pan error to determine the Δ_α and Δ_γ intercepts of the constant error curves.

To determine the error in each case, we again use a first-order Taylor series expansion. For x_e this is

$$x_e = \frac{\partial x_{t-1}}{\partial \alpha}\Delta_\alpha + \frac{\partial x_{t-1}}{\partial \gamma}\Delta_\gamma \tag{7.18}$$

Since $\Delta_\gamma = 0$ for the Δ_α axis intercept, we simplify this as

$$x_e = \frac{\partial x_{t-1}}{\partial \alpha}\Delta_\alpha = f\frac{x_t^2 + \gamma y_t f + f^2}{(-\alpha x_t + \gamma y_t + f)^2} \tag{7.19}$$

and similarly

$$y_e = \frac{\partial y_{t-1}}{\partial \gamma}\Delta_\gamma = f\frac{\alpha x_t f - y_t^2 - f^2}{(-\alpha x_t + \gamma y_t + f)^2} \tag{7.20}$$

For the worst-case error, $(x_t, y_t) = (255, -255)$. The angles of rotation were set to $\alpha = 5°, \gamma = 5°$. Since the assumption of our system equations is that $\sin\alpha \approx \alpha$, and similarly for γ, 5° is a good cut-off point for this approximation and thus gives us the limits of the worst-case angles of rotation.

x_e in pixels	Δ_α in degrees	y_e in pixels	Δ_γ in degrees
1	0.054961	1	0.054961
2	0.109924	2	0.109924
3	0.164886	3	0.164886
4	0.219849	4	0.219849
5	0.274890	5	0.274890
6	0.329772	6	0.329772
7	0.384734	7	0.384734
8	0.439696	8	0.439696
9	0.494658	9	0.494658
10	0.549620	10	0.549620
11	0.604581	11	0.604581

Table 7.2: Worst-case compensation error

Solving for Δ_α and Δ_γ in equations (7.19) and (7.20) we generated Table 7.2. The magnitude of the angle error for given x_e and y_e are the same, which implies we have a circle again, but with slightly smaller radii. This signifies less required angle error for a given error in pixel mapping.

7.4 Significance of error analysis

As shown in Sections 7.2 and 7.3, the pixel mapping error is linearly dependent upon the magnitude of the error in angle information ($\sqrt{{\Delta_\alpha}^2 + {\Delta_\gamma}^2}$). In this section we investiage the consequences of this error and the constraints it places on our system.

7.4.1 Maximum speed of tracking

We assume that the primary source of angle error in a real-time implementation is due to synchronization error, as defined in Section 5.4. For a fixed filtering strategy we can determine the upper bound on the speed or rotation for our system and thus the maximal angular velocity of a target that can be successfully

tracked.

For rotation at a angular velocity of ω_{max}, the angular error caused by poor synchronization will be

$$\theta_e = \omega_{max}\Delta t \tag{7.21}$$

where Δt is the error in timing. Since compensation error is linearly dependent upon angular position error, the compensation error is

$$e = K\theta_e$$

where K is a constant determined by the system parameters. In the example given in Section 7.3, $K = 1/0.054961$.

For a morphological mask of size $n \times n$, the error tolerance will be n. That is to say, if the compensation error is greater than n, the noise caused by this error will not be removed. If the error is less than or equal to n, the noise will be removed. Thus, for the boundary condition,

$$n = K\theta_e \tag{7.22}$$

Substituting equation 7.21 into equation 7.22 and solving for ω_{max} we obtain

$$\omega_{max} = \frac{n}{K\Delta t}$$

We can see that as synchronization error increases, the maximum possible angular velocity decreases. Yet as the size of the morphological filter, n, increases, so does ω_{max}.

7.4.2 Minimum speed of tracking

For a moving target with a very slow angular velocity relative to the camera, it is possible that the target will not be detected, since any motion caused by it will be removed with the morphological filtering. If we consider a target moving at the slowest detectable speed, ω_{min}, the angle covered by this target each sample instant, t_s, will be

$$\theta_{min} = \omega_{min}t_s \tag{7.23}$$

The distance on the image plane this will move can be calculated by

$$d = f \tan \theta_{min} \tag{7.24}$$

thus

$$\theta_{min} = \arctan \frac{d}{f} \tag{7.25}$$

If we are using an $n \times n$ filter mask, the object must move a minimum of $n + 1$ pixels to be identified, and

$$\theta_{min} = \arctan \frac{n+1}{f} \tag{7.26}$$

Substituting equation (7.23) into equation (7.26) and solving for ω_{min} yields

$$\omega_{min} = \frac{\arctan \frac{n+1}{f}}{ts} \tag{7.27}$$

Thus, to detect a slow moving object it is desirable to either decrease the filter size n, or increase the sample time t_s.

7.4.3 Filtering and sampling strategy

As we can see, the desirable filter size is not identical for different moving objects. Ideally, we would like to set the filter size according to the camera motion and the estimated motion of the target. Initially, before any target is acquired, the camera may remain stationary with no filtering required, or conduct a slow search path which would minimize compensation error, and the chance of missing a target. As an object is tracked, and the angular velocity is estimated and predicted, the optimal filtering solution could be determined. The difficulty with this *adaptable* filtering strategy, is that to implement different sized filters on a constantly changing basis is demanding on the hardware and not realistically implementable on most pipeline image-processing boards.

Chapter 8

Conclusion

8.1 Summary

The objective of this thesis is to design methods of tracking moving objects with a pan/tilt camera for real-time implementation.

It was shown that for a camera constrained to rotation, identical scene information can be extraced from different camera positions. This allows the dvelopment of a mapping relationship between images taken at different camera orientations. With images compensated for camera rotations, static camera techniques can be applied to active camera image sequences.

Since compensation is susceptible to errors caused by poor camera position information, morphological filters are employed to remove erroneously detected motion. While this method successfully removes false motion, large filter masks impose an additional computational burden on the system and must be intelligently selected.

Improving camera position information greatly relieves the filtering requirements and is necessary for a real-time implementation.

8.2 Assessment

Although active camera systems have many theoretical benefits, at present there is still much work to be done before the many additional problems they impose can

be satisfactorily solved. The method presented here is computationally fast given appropriate hardware and accurate inputs. However, for improved performance, additional strategies, such as variable filtering, as well as customized hardware specific for compensation and morphological filtering should be developed.

The system demonstrated yields reasonable results, considering the unconstrained and cluttered background, and the arbitrariness of the moving object.

8.3 Future research

As yet, this system has not been implemented on a real-time platform. With the availability of pipline-architecture image-processing hardware, even without customization, the methods presented here can be implemented in real-time.

With the basic system implemented, there are several possible modifications which can be made to expand and improve the performance of the system.

As mentioned in Section 7.4.3, with a thorough knowledge of the errors present in a system, the performance could be improved using an adaptible fitering strategy. As well, detecting texture information about the general background would enable the system to select the most appropriate motion detection techniques. An independent ranging technique, such as focus-ranging, makes it possible to obtain 3-D information about the tracked object. This would improve the scope of th system and opens up possibilities for sensor fusion with robotics systems.

As discussed in Section 3.3.2, there are computer vision techniques which require an active camera to operate. Techniques such as variable resolution could be implemented and tested with a real-time pan/tilt tracking system.

Since within all the basic equations and derivations of this work, the focal length plays a key role, it would be interesting to investigate the potential with an active zoom in which the focal length could be changed. For instance, for erratically moving objects, a wide-angle lens would provide more robust tracking, whereas for a predictable object, a telephoto lens should improve the position estimates.

The use of color images would enhance the motion detection by adding additional solutions to the contrast problem common with image subtraction tech-

niques.

Bibliography

[1] Peter Allen, Billibon Yoshimi, and Aleksander Timcenko. Real-time visual servoing. Technical Report CUCS-035-90, Columbia University, Pittsburgh, PA, USA, September 1990.

[2] Peter K. Allen. Real-time motion tracking using spatio-temporal filters. In *Proceedings of the DARPA Image understanding workshop*, pages 695–701, Palo Alto, California, USA, May 1989.

[3] John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active vision. In *Proceedings - First International Conference on Computer Vision*, pages 35–54, London, England, June 1987.

[4] Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):996–1005, 1988.

[5] Dana H. Ballard and Christopher M. Brown. *Computer Vision.* Prentice-Hall Inc., 1982.

[6] Anup Basu and Xiaobo Li. A frame-work for variable resolution. In *Advances in Computing and Information - ICCI '91*, pages 721–732, Ottawa, Canada, May 1991.

[7] Anup Basu and Xiaobo Li. Variable-resolution character thinning. *Pattern Recognition*, 12(4):241–248, 1991.

[8] Anup Basu and Sergio Licardie. Variable resolution vergence control. Technical report, University of Alberta, Computing Science Dept., 1992.

[9] Bir Bhanu and Wilhelm Burger. Qualitative understanding of scene dynamics for moving robots. *International Journal of Robotics Research*, 9(6):74–90, 1990.

[10] Alistair J. Bray. Tracking ojects using image disparities. *Image and Vision Computing*, 8(1):4–9, February 1990.

[11] Rodney Allen Brooks. *Model-based computer vision*. UMI Research Press, 1984.

[12] Christopher Brown, Hugh Durrant-Whyte, John Leonard, and Bobby Rao. Centralized and decentralized kalman filter techniques for tracking, navigation, and control. In *Proceedings - Image understanding workshop*, pages 651–675, Palo Alto, California, USA, May 1989.

[13] James H. Duncan and Tsai-Chia Chou. Temporal edges: The detection of motion and the computation of optical flow. In *Second Int Conf on Computer Vision*, pages 374–382, Tampa, FL, USA, December 1988.

[14] T. S. Huang and Yen B. L. Determining 3-d motion and structure of a rigid body using straight line conrrespondences. In *Proceedings of the NATO advanced study institute on image sequence processing and dynamic scene analysis*, pages 365–393, Braunlage, Federal Republic of Germany, June 1982.

[15] Paul Kalata. The tracking index: a generalized parameter for $\alpha - \beta$ and $\alpha - \beta - \gamma$ trackers. *IEEE Transactions on Aerospace and Electronic Systems*, AES-20(2):174–182, 1984.

[16] Ken-ichi Kanatani. Camera rotation invariance of image characteristics. *Computer Vision, Graphics, and Image Processing*, 39(3):328–354, September 1987.

[17] Ken-ichi Kanatani. Coordinate rotation invariance of image characteristics for 3d shape and motion recovery. In *Proceedings - First International Conference on Computer Vision*, pages 55–64, London, England, June 1987.

[18] Youngchul Kay and Lee Sukhan. A robust 3-d motion estimation with stereo cameras on a robot manipulator. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 1102–1107, Sacramento, California, USA, April 1991.

[19] David G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[20] Petros Maragos. Tutorial on advances in morphological image processing and analysis. *Optical Engineering*, 26(7):623–32, July 1987.

[21] Randal C. Nelson. Qualitative detection of motion by a moving observer. In *Proceedings - IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 173–178, Maui, Hawaii, USA, June 1991.

[22] P. D. Picton. Tracking and segmentation of moving objects in a scene. In *Third International Conference on Image Processing and its Applications*, pages 389–393, Coventry, Engl, 1989.

[23] Azriel Rosenfeld and Rama Chellappa. Computer vision : Attitudes, barriers, counseling. *Proceedings of Vision Interface '92*, pages 1–7, May 1992.

[24] R. S. Stephens. Real-time 3d object tracking. *Image and Vision Computing*, 8(1):91–96, 1990.

[25] Massimo Tistarelli and Guilio Sandini. Robot naviation using an anthropomorphic visual sensor. In *Proceedings - 1990 IEEE International Conference on Robotics and Automation*, pages 374–381, Cincinnati, Ohio, USA, May 1990.

Appendix A

Projection of a sphere onto the image plane

This analysis discusses the relationship of the 3-D centroid of a sphere projected onto the image plane and the 2-D centroid of the image of that sphere as presented in Section 3.3.2. However, before we can determine the 2-D centroid of the image, we must derive the shape of the image. The projection of a sphere onto the image plane will be the same as that of a circle formed by the occluded volume of the sphere. The sphere has a radius r_s and range to the origin R_s while the occluded circle has a radius r_c and a range to the origin R_c. The plane of the circle will be perpendicular to a line from the origin to the centre of the circle.

The boundary of the circle is formed by the points where lines that intersect the origin lie tangent to the surface of the sphere. In Figure A.1 we can see the geometry of the problem described. If the dimensions of the sphere and the range to the centre of the sphere are known, from similar triangles we obtain

$$sin(\gamma) = \frac{r_s}{R_s}$$

from which, using the identity $cos^2 + sin^2 = 1$ we find

$$cos(\gamma) = \sqrt{1 - \frac{r_s^2}{R_s^2}}$$

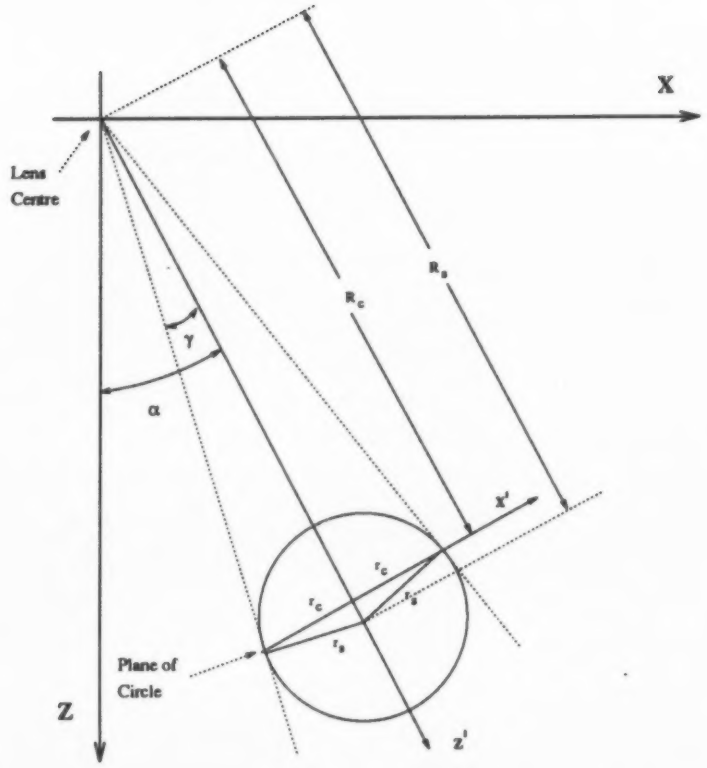


Figure A.1:
R_s – distance between focal point and centroid of sphere
R_c – distance between focal point and centre of circle seen by the camera
r_s – radius of the sphere
r_c – radius of the circle seen by the camera

By considering the geometry presented in Figure A.1 we derive

$$r_c = r_s \cos(\gamma) = r_s \sqrt{1 - \frac{r_s^2}{R_s^2}}$$

$$R_c = R_s - r_s \sin(\gamma) = R_s(1 - \frac{r_s^2}{R_s^2})$$

Now that we have determined the information defining the circle, let us consider its projection on the image plane. From the symmetry of the geometry of the imaging process it follows that we can rotate the coordinate frame about the Z-axis without change in the image shape. Therefore, without loss of generality, we consider a particular case for which the centre of the sphere lies on the X-Z plane (i.e. $Y_c = 0$). We then create a new coordinate frame at the centre of the circle, shown in Figure A.1. By projecting the new coordinate frame onto the world frame, we obtain the relationship between the two frames as

$$X' = (X - X_c)\cos(\alpha) + (Z - Z_c)\sin(\alpha) \tag{A.1}$$

$$Y' = Y \tag{A.2}$$

$$Z' = (Z - Z_c)\cos(\alpha) - (X - X_c)\sin(\alpha) \tag{A.3}$$

Since

$$X_c = R_c \sin(\alpha) \tag{A.4}$$

$$Z_c = R_c \cos(\alpha) \tag{A.5}$$

by substituting equations (A.4) and (A.5) into (A.1) and (A.3) yields

$$X' = X\cos(\alpha) + Z\sin(\alpha) - 2R_c\cos(\alpha)\sin(\alpha) \tag{A.6}$$

$$Y' = Y \tag{A.7}$$

$$Z' = Z\cos(\alpha) - X\sin(\alpha) - R_c\cos(2\alpha) \tag{A.8}$$

The circle of interest is described in the new coordinate frame by the equation of a circle with centre at the origin. Hence

$$r_c^2 = (X')^2 + (Y')^2$$

or, by substituting in equations (A.6)

$$r_c^2 = (X\,cos(\alpha) + Z\,sin(\alpha) - 2R_c\,cos(\alpha)sin(\alpha))^2 + Y^2 \tag{A.9}$$

The condition of equation (A.9) is that of the equation of the circle to be projected on the image plane. If we recall the relationship between points in three dimensions and their projections on the image place (see Section 3.2.2)

$$x = \frac{fX}{Z}, \quad y = \frac{fY}{Z} \tag{A.10}$$

it follows that

$$X = \frac{xZ}{f}, \quad Y = \frac{yZ}{f} \tag{A.11}$$

Using equation (A.11), equation (A.9) can be modified to

$$r_c^2 = (\frac{Zx\cos(\alpha)}{f} + Z\sin(\alpha) - 2R_c\cos(\alpha)\sin(\alpha))^2 + (\frac{Zy}{f})^2 \tag{A.12}$$

Now we have reduced the equation to terms of only (x, y) and Z. To eliminate the Z terms, we will utilize the fact that the circle in question lies on a plane which we can describe by

$$Z = -X\,\frac{sin(\alpha)}{cos(\alpha)} + \frac{R_c}{cos(\alpha)} \tag{A.13}$$

Again, using equations (A.10) this reduces to

$$Z = \frac{fR_c}{f\cos(\alpha) + x\sin(\alpha)} \tag{A.14}$$

By substituting equation (A.14) into equation (A.12) we obtain

$$r_c^2 = R_c^2\frac{[(x\,cos(\alpha) - f\,sin(\alpha))^2cos^2(2\alpha) + y^2]}{(f\,cos(\alpha) + x\sin(\alpha))^2} \tag{A.15}$$

This equation describes the image created by the projection of the sphere onto the image plane in terms of constants of the system and in the image plane coordinates.